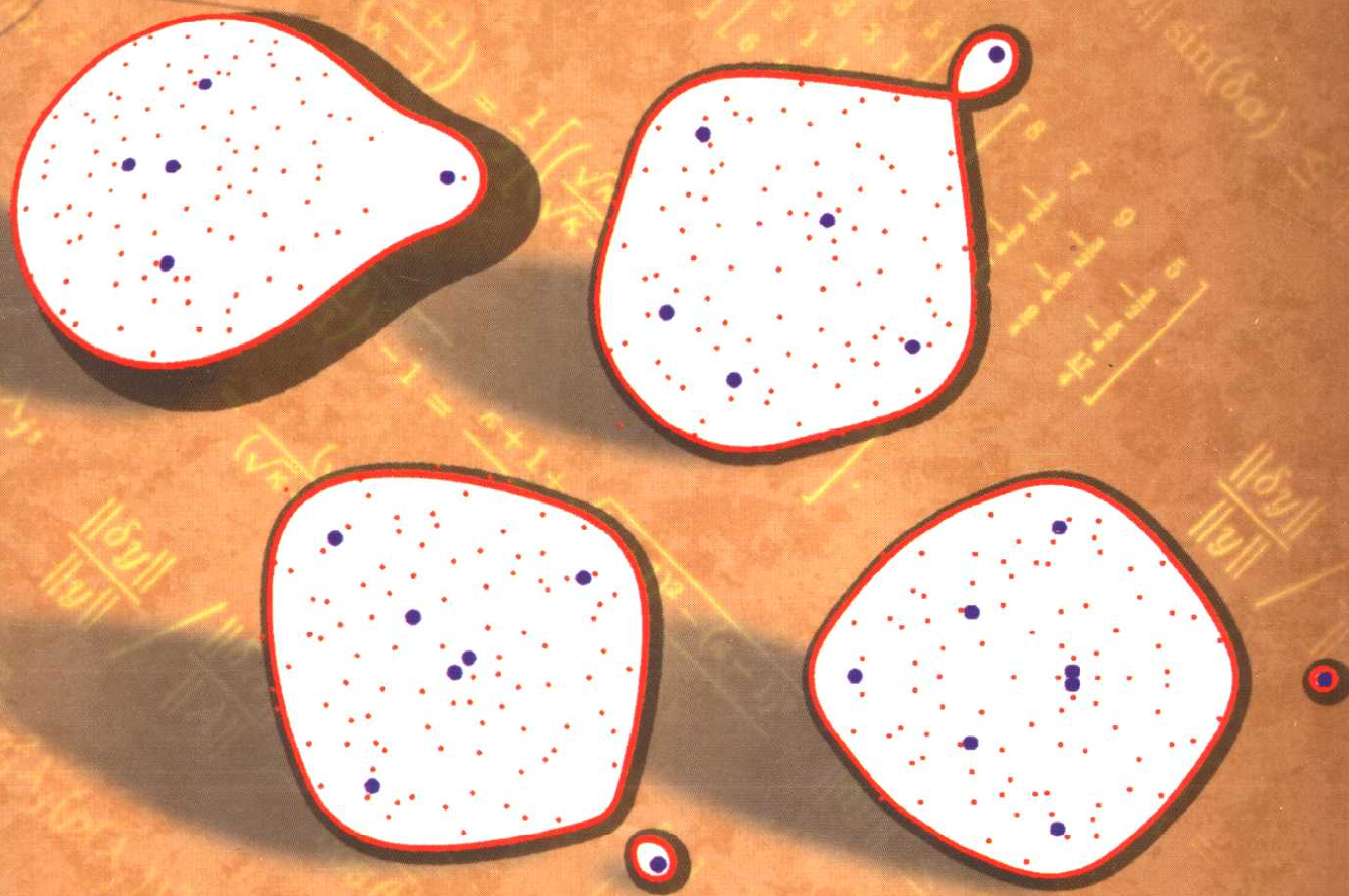
TURING 图灵数学·统计学丛书

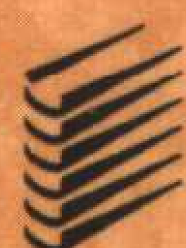


Numerical Linear Algebra

数值线性代数

Lloyd N. Trefethen
David Bau, III 著

陆金甫 关治 译


人民邮电出版社
POSTS & TELECOM PRESS

数值线性代数

Numerical Linear Algebra

“这是一部优秀的教科书，书中的论述独特且富有创造性，必将使从事本领域教学的所有人受益。”

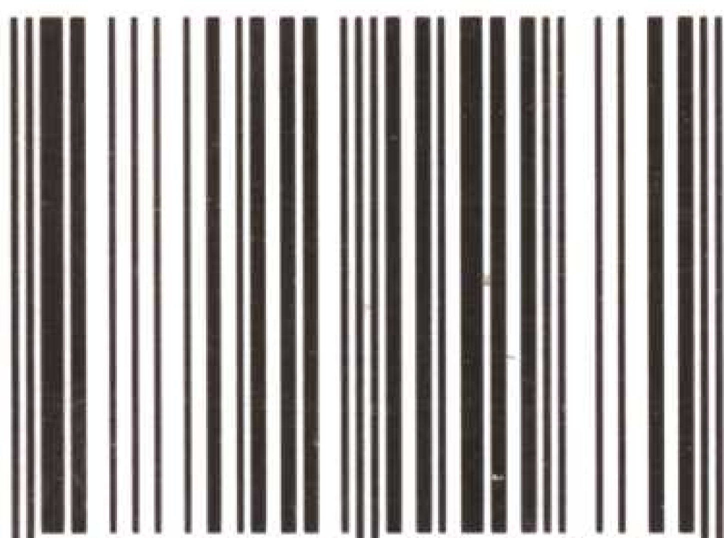
——曼彻斯特大学应用数学教授 Nicholas J. Higham

本书是一本数值线性代数领域的导论性著作，适合作为研究生或高年级本科生课程的教材。作者深刻阐述了数值线性代数的基本思想，加上他们清新、引人入胜的写作风格，使得本书成为本领域的经典之作。

Lloyd N. Trefethen 是世界顶尖的数值分析学家。英国牛津大学教授，该校数值分析研究组的负责人。曾任教于 MIT, Cornell 及纽约大学柯朗研究所。Trefethen 还是英国皇家学会院士，因其在数值分析领域杰出的成就获第一届 Fox 奖。**David Bau, III** 是 Trefethen 在 Cornell 大学的研究生，他目前作为一个软件研发人员在微软公司互联网部门工作。

本书相关信息请访问：图灵网站 http://www.turingbook.com
读者/作者热线：(010) 88593802
反馈/投稿/推荐信箱：contact@turingbook.com

上架建议 计算数学



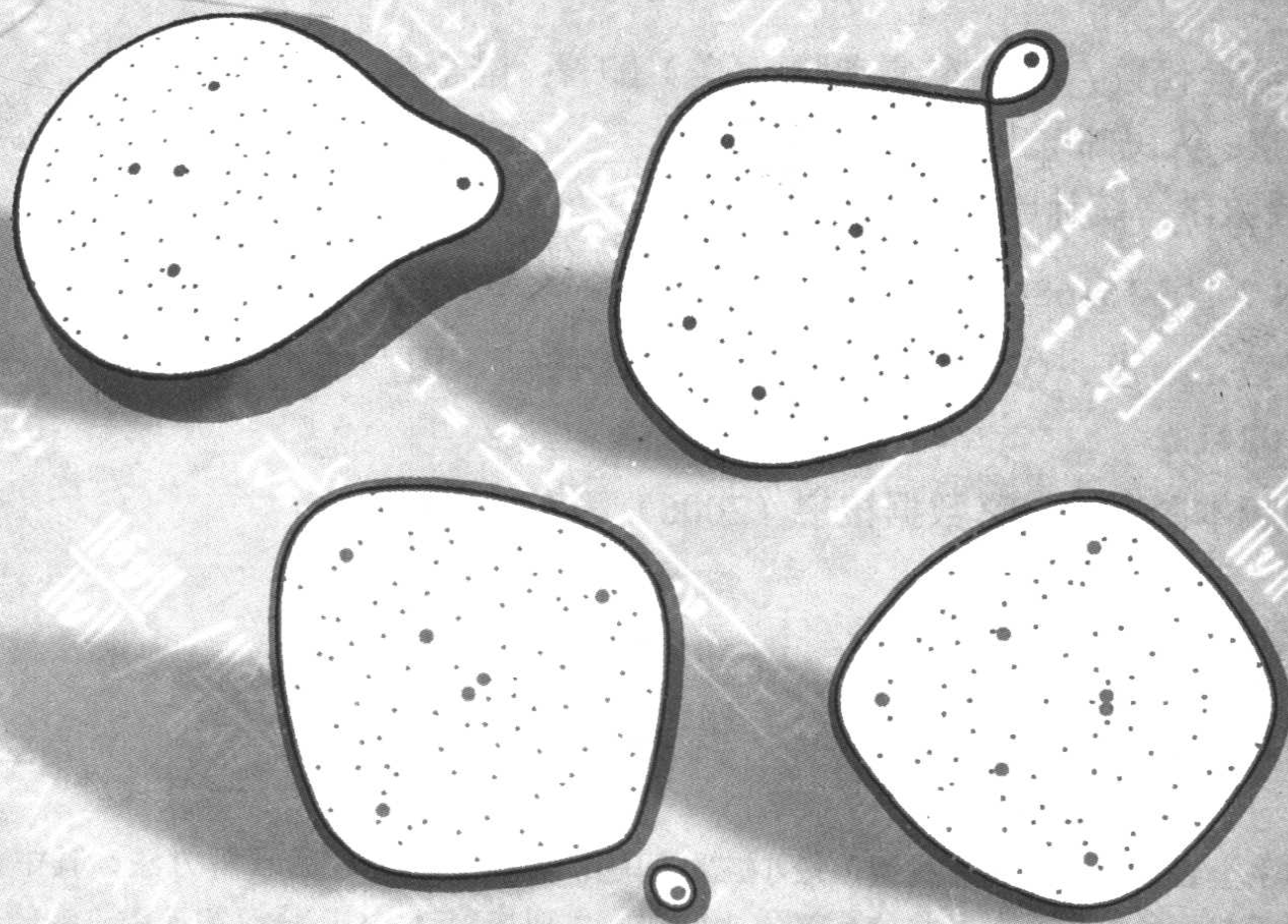
ISBN 7-115-15168-7
9 787115 151681 >

ISBN7-115-15168-7/O1・3

定价：39.00 元

人民邮电出版社网址 www.ptpress.com.cn

TURING 图灵数学·统计学丛书



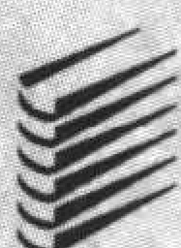


Numerical Linear Algebra

数值线性代数

Lloyd N. Trefethen
David Bau, III 著

陆金甫 关治 译



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目（CIP）数据

数值线性代数 /（英）特雷弗腾，（美）鲍著；陆金甫，关治译.
—北京：人民邮电出版社，2006.11
（图灵数学·统计学丛书）

ISBN 7-115-15168-7

Ⅰ. 数… Ⅱ. ①特…②鲍…③陆…④关… Ⅲ. 线性代数计算法－教材
Ⅳ. 0241.6

中国版本图书馆 CIP 数据核字（2006）第 098728 号

内 容 提 要

本书全面论述了线性方程组、最小二乘问题以及特征值问题的求解方法，其中包含不少新近发展起来的方法. 全书共分 6 部分，40 讲。主要内容有：QR 分解和最小二乘问题、条件与稳定性、求解线性方程组的直接方法、特征值问题及迭代方法.

本书可作为计算数学专业、科学和工程学科高年级本科生或研究生一学期的教材，也可供应用工作者参考.

图灵数学·统计学丛书

数值线性代数

◆ 著　　　Lloyd N.Trefethen　David Bau, III
译　　　陆金甫　关　治
责任编辑　武卫东

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京铭成印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：700 × 1000　1/16
印张：20.5
字数：425 千字　　2006 年 11 月第 1 版
印数：1－4 000 册　　2006 年 11 月北京第 1 次印刷

著作权合同登记号　图字：01-2006-1278 号

ISBN 7-115-15168-7/O1 · 3

定价：39.00 元

读者服务热线：(010)88593802　印装质量热线：(010)67129223

版权声明

符　　号

对正方形或矩形矩阵 $\boldsymbol{A} \in \mathbb{C}^{m\times n}$，$m \geqslant n$：

QR 因子分解：$\boldsymbol{A} = \boldsymbol{QR}$

约化 QR 因子分解：$\boldsymbol{A} = \hat{\boldsymbol{Q}}\hat{\boldsymbol{R}}$

SVD：$\boldsymbol{A} = \boldsymbol{U}\Sigma\boldsymbol{V}^*$

约化 SVD：$\boldsymbol{A} = \hat{\boldsymbol{U}}\hat{\Sigma}\boldsymbol{V}^*$

对正方形矩阵 $\boldsymbol{A} \in \mathbb{C}^{m\times m}$：

LU 因子分解：$\boldsymbol{PA} = \boldsymbol{LU}$

楚列斯基因子分解：$\boldsymbol{A} = \boldsymbol{R}^*\boldsymbol{R}$

特征值分解：$\boldsymbol{A} = \boldsymbol{X}\Lambda\boldsymbol{X}^{-1}$

舒尔因子分解：$\boldsymbol{A} = \boldsymbol{UTU}^*$

正交投影算子：$\boldsymbol{P} = \hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^*$

豪斯霍尔德镜射算子：$\boldsymbol{F} = \boldsymbol{I} - 2\dfrac{\boldsymbol{vv}^*}{\boldsymbol{v}^*\boldsymbol{v}}$

QR 算法：$\boldsymbol{A}^k = \underline{\boldsymbol{Q}}^{(k)}\underline{\boldsymbol{R}}^{(k)}$，　$\boldsymbol{A}^{(k)} = (\underline{\boldsymbol{Q}}^{(k)})^{\mathrm{T}}\boldsymbol{A}\,\underline{\boldsymbol{Q}}^{(k)}$

阿诺尔迪迭代：$\boldsymbol{AQ}_n = \boldsymbol{Q}_{n+1}\widetilde{\boldsymbol{H}}_n$，　$\boldsymbol{H}_n = \boldsymbol{Q}_n^*\boldsymbol{AQ}_n$

兰乔斯迭代：$\boldsymbol{AQ}_n = \boldsymbol{Q}_{n+1}\widetilde{\boldsymbol{T}}_n$，　$\boldsymbol{T}_n = \boldsymbol{Q}_n^{\mathrm{T}}\boldsymbol{AQ}_n$

译者介绍

陆金甫　清华大学数学科学系教授．研究方向为偏微分方程差分方法及并行计算．长期从事计算数学的教学与研究，参加和负责过多次国家自然科学基金项目．

关　治　清华大学数学科学系教授，专业方向为计算数学．长期从事基础数学与计算数学的教学与研究，编著《全国工程硕士专业学位入学资格考试考前辅导教程》及其他多部教材与专著，曾多次获清华大学优秀教学成果奖．

两译者曾合作编著了《数值分析基础》（高等教育出版社）、《偏微分方程数值方法》（清华大学出版社）等教材．

译 者 序

L. N. Trefethen 和 D. Bau, Ⅲ的《数值线性代数》是一本有特色的教材. 它适用于计算数学专业高年级本科生或研究生课程，也是科学与工程学科研究生的一本好的参考书.

本书在不大的篇幅中，全面论述了线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$、最小二乘问题 $\min\limits_{x}\|\boldsymbol{b}-\boldsymbol{Ax}\|_2$ 以及特征值问题的求解方法，包括了基本求解方法及新近发展起来的新方法，全书分为基础知识、QR 分解和最小二乘法问题、条件与稳定性、求解线性方程组的直接方法、特征值问题以及迭代方法 6 个部分. 特征值问题求解及迭代方法是本书的重点，它们占了本书近一半篇幅. 全书共 40 讲，每讲讲述 1 个或 2 个主题，安排内容都不是太多，便于课堂讲授.

与通常的数值线性代数教材相比较，该教材具有显著特点. 在内容安排上，把多处用到的 QR 分解方法放在高斯消元法前面讲述；在迭代方法的材料选择上基本不涉及经典的方法，而几乎全部讲述克雷洛夫子空间迭代方法；在算法的讲述上特别注重算法的思想，强调了各个算法之间的联系，例如共轭梯度法与多项式逼近，兰乔斯迭代与高斯求积方法等；几何方面的说明在教材中起到了重要作用，例如在阿诺尔迪的特征值方法中采用了“阿诺尔迪双纽线”来说明方法的收敛性等等.

本书正文后有一个附录《数值分析的定义》，它是第一作者已发表过的一篇文章. 也已译出，供参考.

本教材中有些方法细节未详细讲述，有关内容，读者可参考 J. W. Demmel 的 *Applied Numerical Linear Algebra*（1997）[1]和 Y. Saad 的 *Iterative Methods for Sparse Linear Systems*（2nd edition, 2003）等书.

本书从开头到第 23 讲由关治翻译，其后部分由陆金甫翻译，张宝琳研究员抽空看过部分译稿并提出了很好的意见.

由于译者的英文水平和学科水平都有限. 译文难免有不妥之处，请广大读者批评指正.

译 者

2006 年 4 月

1. 该书的中译本《应用数值线性代数》也已由人民邮电出版社出版. ——编者注

前　言

从上个世纪 80 年代初期开始，本书第一作者在麻省理工学院和康奈尔大学讲授数值线性代数的研究生课程. 听过这门课程的学生如今已有数以百计之多，他们都是来自工程和物理科学等相关领域的研究生. 本书就是在这门课程所讲述的内容的基础上整理而成的.

在数值线性代数领域，已经有一本百科全书式的著作，即 Golub 和 Van Loan 的 *Matrix Computations*[1]，现已有第三版. 本书绝不是要重复前人的工作，而是想以尽可能简练的方式给出基本的数学思想. 本书篇幅小，适合于一学期教学之用. 当然，我们仍希望本书的每一位读者阅读 Golub 和 Van Loan 的书，了解进一步的细节和其他相关主题，以及书中为该领域研究提供的丰富的参考文献. 最近新出了两本重要的书，由 Higham 和 Demmel 所著，我们将在本书最后的注记中提到.

数值线性代数，虽然名字略显沉闷，但是这一领域本身却是多彩且基础性的. 说它多彩，是因为它充满了极富生命力的思想，这完全不同于在数学系线性代数课程通常所强调的内容.（在每学期结束之时，学生们总是说这门课程要比他们一直想像的有更多的内涵.）说它是基础性的，是因为由于历史原因“数值”线性代数事实上就是指应用线性代数. 这里涵盖了每个数学家有效地应用向量和矩阵都需要的基本思想. 事实上，我们的课题并不仅仅局限于向量和矩阵，因为我们做的每一件事情都涉及函数和算子. 数值线性代数在真正意义上归于泛函分析，只不过它的重点总是在实际的算法思想而不是数学上的技术细节.

本书共分 40 讲. 我们力图使每讲围绕一个或两个中心思想，强调各主题之间的统一性并且不拘泥于细节的处理. 许多地方的处理方式并非是标准的. 这里不便一一列出（参看注记），但是我们会指出本书中一个比较特殊的地方. 我们将一反惯例，不从高斯消元法开始，那是数值线性代数中非典型的算法，极其难以分析，而对每个初学本课程的学生，它又过于熟知因而有些乏味. 代之的是，我们从 QR 因子分解开始，它更重要，而且通俗易懂，对大多数学生来说是新鲜的思想. QR 因子分解是联系数值线性代数大多数算法的纽带，包括最小二乘、特征值和奇异值问题等方法，同时也包括对这些问题和对方程组的迭代方法. 自从上个世纪 70 年代以来，迭代方法已经占据了科学计算的中心地位，对它们，我们安排在本书的最后一部分加以介绍.

我们希望读者能同意我们如下的见解，即如果除了微积分与微分方程之外，还有什么数学领域是数学科学的基础的话，那就是数值线性代数.

---

1. 中译本为：《矩阵计算》，袁亚湘等译，科学出版社，2001. ——译者注

致　谢

如果没有众多人士的帮助，我们不可能完成此书. 首先要感谢修读麻省理工学院课程 Math 335 和康奈尔大学课程 CS 621 的数以百计的研究生，他们在整个 10 年期间的热情和建议影响了本课程内容的选择和表达的方式. 在康奈尔大学，这些学生中大约有 70 人使用了本书的初稿，他们提出了很多有益的建议. Keith Sollers 就独自指出了大量排印错误.

Trefethen 的大多数研究生在本书写作期间，从头到尾通读了这本教科书——有时是在很短时间内完成的. 感谢 Jeff Baggett、Toby Driscoll、Vicki Howle、Gudbjorn Jonsson、Kim Toh 和 Divakar Viswanath 的很多建设性建议. 拥有众多学生和同事，真是荣幸.

与 SIAM 的出版工作人员的合作是令人愉快的，很少能有出版单位像 SIAM 这样将灵活性和专业性集于一身. 我们十分感谢 SIAM 的编辑、生产和设计人员，他们的共同努力大大增加了本书的吸引力，尤其要感谢 Beth Gallagher，她除了出色地完成了文字编辑工作之外，还自始至终在支持着我们.

康奈尔大学计算机科学系对数值线性代数及其教材出版给予的支持无与伦比. 系里其他 3 位在此方面有兴趣的同事是 Tom Coleman，Charlie Van Loan 和 Steve Vavasis，我们感谢他们在使康奈尔大学成为富于吸引力的科学计算中心的过程中所做的贡献. Vavasis 阅读了本书整个的初稿并提出很多有价值的建议，Van Loan 是最初介绍 Trefethen 来到康奈尔的人. 在我们的非数值方面的同事之中，感谢 Dexter Kozen，他的 *The Design and Analysis of Algorithms* 一书是本书仿效的对象，它也是一本由 40 讲组成的书. 感谢 Rebekah Personius 先生，作为本系的一名行政人员，他的专业特长、勤勤恳恳和良好的工作态度给我们提供了很大的支持.

我们的另一位同事，经常到康奈尔大学访问的学者 Anne Greenbaum 为本书提供了大量建议，她是我们认识的对数值线性代数思考最深入的学者之一.

从 1995 年 9 月到 1996 年 12 月，我们的几位同事用本书的初稿讲授课程，他们提出了自己的意见，也转达了学生们的建议. 其中有 Gene Golub（斯坦福大学）、Bob Lynch（普度大学）、Suely Oliveira（得克萨斯 A & M 大学）、Michael Overton（纽约大学）、Haesun Park 和 Ahmed Sameh（明尼苏达大学）、Irwin Pressmann（加拿大卡尔顿大学）、Bob Russell 和 Manfred Trummer（加拿大西蒙・费雷泽大学）以及 Hong Zhang 和 Bill Moss（美国克莱姆森大学）. 在这些人中打破纪录的是 Lynch 和 Overton，他们每人都提出了长长的一系列的详细建议. 虽然我们写书时已尽量一丝不苟，但他们的意见还是非常合理的，不能忽略，而现在本书中上百处改进正是因为 Lynch 或 Overton 的帮助.

要说到有什么重要的帮助使本书更加完善，我们要着重提及曼彻斯特大学的Nick Higham教授，他所提供的帮助（虽然他不愿意承让）是我们永远的情债，他的创造性和细致学风启迪了一批比他年幼或年长的数值分析学家（最小的是他年龄的一半，最大的是他年龄的两倍）．虽然时间很短，Higham却带着他特有的好意精心地阅读了本书的初稿并提出了很多页技术上的建议，其中某些建议深深地改变了本书．

几十年来，数值线性代数已经成为充满友爱和富有凝聚力的领域的一种典范．Trefethen要特别感谢3位“教父级的长者”，是他们的课堂讲授首先将他吸引到这个主题，他们是Gene Golub，Cleve Moler和Jim Wilkinson．

除了数值线性代数之外，还有更多事情使生命更有意义．为此，第一作者要感谢家人Anne、Emma(5岁)和Jacob(3岁)，第二作者要感谢Heidi Yeh．

目 录

第 I 部分

基　　础

- 第 1 讲　矩阵-向量乘法
- 第 2 讲　正交向量和矩阵
- 第 3 讲　范数
- 第 4 讲　奇异值分解
- 第 5 讲　SVD 的进一步讨论

第 1 讲　矩阵-向量乘法

我们已经知道了矩阵-向量乘法的公式，然而，第 1 讲的目的是描述一种解释这种乘积的方法，它可能是人们不太熟悉的. 如果 $\boldsymbol{b}=\boldsymbol{A}\boldsymbol{x}$，则 $\boldsymbol{b}$ 是 $\boldsymbol{A}$ 的列的一个线性组合.

1.1　熟知的定义

令 $\boldsymbol{x}$ 为一个 n 维列向量，$\boldsymbol{A}$ 为一个 $m\times n$ 矩阵（m 行 n 列）. 则矩阵-向量乘积 $\boldsymbol{b}=\boldsymbol{A}\boldsymbol{x}$是如下定义的 m 维列向量：

$$b_i=\sum_{j=1}^{n}a_{ij}x_j,\qquad i=1,\cdots,m. \tag{1.1}$$

这里，b_i 记为 b 的第 i 个元素，a_{ij}记为 A 的 i,j 元素（第 i 行第 j 列），x_j 记为 $\boldsymbol{x}$ 的第 j 个元素. 为了简化，在本书中除了几讲外，都假设这些量属于复数域$\mathbb{C}$. m 维向量空间为$\mathbb{C}^m$，$m\times n$ 矩阵的空间为$\mathbb{C}^{m\times n}$.

映射 $\boldsymbol{x}\mapsto\boldsymbol{A}\boldsymbol{x}$ 是线性的（linear），意思是对任意的 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{C}^n$ 和任意的 $\alpha\in\mathbb{C}$，

$$\begin{aligned}\boldsymbol{A}(\boldsymbol{x}+\boldsymbol{y}) &= \boldsymbol{A}\boldsymbol{x}+\boldsymbol{A}\boldsymbol{y},\\ \boldsymbol{A}(\alpha\boldsymbol{x}) &= \alpha\boldsymbol{A}\boldsymbol{x}.\end{aligned}$$

反之，每个由$\mathbb{C}^n$ 到$\mathbb{C}^m$ 的线性映射都对应一个 $m\times n$ 矩阵.

1.2　矩阵乘向量

令 m 维向量 $\boldsymbol{a}_j$ 记为 $\boldsymbol{A}$ 的第 j 列，那么（1.1）可重写成

$$\boldsymbol{b}=\boldsymbol{A}\boldsymbol{x}=\sum_{j=1}^{n}x_j\boldsymbol{a}_j. \tag{1.2}$$

这个方程可以如下格式显示：

$$\left[\begin{array}{c} \\ \boldsymbol{b} \\ \\ \end{array}\right]=\left[\begin{array}{c|c|c|c} & & & \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_n \\ & & & \end{array}\right]\left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array}\right]=x_1\left[\begin{array}{c} \\ \boldsymbol{a}_1 \\ \\ \end{array}\right]+x_2\left[\begin{array}{c} \\ \boldsymbol{a}_2 \\ \\ \end{array}\right]+\cdots+x_n\left[\begin{array}{c} \\ \boldsymbol{a}_n \\ \\ \end{array}\right].$$

在（1.2）里，$\boldsymbol{b}$ 表示为列 $\boldsymbol{a}_j$ 的线性组合. 由（1.1）到（1.2）只是记号稍作改变，然而，要正确理解数值线性代数算法，理解（1.2）中的 $\boldsymbol{A}\boldsymbol{x}$ 运算形式是非常重要的.

我们可以用下面的方法综合矩阵-向量乘积各种不同的描述. 作为数学家，我们将公式$Ax=b$看成A作用在x上产生b；反之，公式（1.2）则解释为x作用在A上产生b.

例 1.1 范德蒙德矩阵. 固定一个数列$\{x_1,x_2,\cdots,x_m\}$，如果p和q都是次数$<n$的多项式，α是一个标量，那么$p+q$和αp也是次数$<n$的多项式，而且，这些多项式在点x_i的值满足下列线性性质：

$$(p+q)(x_i)=p(x_i)+q(x_i),$$
$$(\alpha p)(x_i)=\alpha(p(x_i)).$$

因此，由次数$<n$的多项式p的系数向量到多项式样本值的向量$(p(x_1),p(x_2),\cdots,p(x_m))$的映射是线性的. 任意一个线性映射可以表示为矩阵的乘法. 这就是一个例子. 事实上，它表示为一个$m\times n$阶范德蒙德矩阵（Vandermonde matrix）

$$A=\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{n-1} \end{bmatrix}.$$

4

如果c是p的系数列向量

$$c=\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{bmatrix},\quad p(x)=c_0+c_1x+c_2x^2+\cdots+c_{n-1}x^{n-1},$$

那么，乘积Ac将给出多项式样本值，即对从1到m的每个i，我们有

$$(Ac)_i=c_0+c_1x_i+c_2x_i^2+\cdots+c_{n-1}x_i^{n-1}=p(x_i). \tag{1.3}$$

在这个例子里，显然矩阵-向量乘积Ac不必像（1.1）所示的那样，看作m个不同的数量和（每个和都给出一个关于c的元素的线性组合），相反，A可以看成一个以单项式的样本值为列的矩阵

$$A=\left[\begin{array}{c|c|c|c|c} & & & & \\ 1 & x & x^2 & \cdots & x^{n-1} \\ & & & & \end{array}\right], \tag{1.4}$$

乘积Ac可以理解为（1.2）形式的单个向量的和，立刻就可以给出这些单项式的线性组合

$$Ac = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1} = p(x).$$ □

本讲余下的部分将用（1.2）的观点重温线性代数的一些基本概念.

1.3　矩阵乘矩阵

对于矩阵-矩阵乘积 $B=AC$，B 的每列是 A 的列的线性组合. 为导出这个事实，我们从通常的矩阵乘积公式出发. 若 A 是 $\ell \times m$ 阶矩阵，C 是 $m \times n$ 阶矩阵，则 B 是 $\ell \times n$ 阶矩阵，其元素定义为

$$b_{ij} = \sum_{k=1}^{m} a_{ik} c_{kj}. \tag{1.5}$$

其中，b_{ij}，a_{ik}和 c_{kj}分别是 B，A 和 C 的元素. 按列来写，此乘积为

$$\left[\begin{array}{c|c|c|c} & & & \\ b_1 & b_2 & \cdots & b_n \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ a_1 & a_2 & \cdots & a_m \\ & & & \end{array}\right] \left[\begin{array}{c|c|c|c} c_1 & c_2 & \cdots & c_n \end{array}\right],$$

5

（1.5）成为

$$b_j = Ac_j = \sum_{k=1}^{m} c_{kj} a_k. \tag{1.6}$$

b_j 是诸列向量 a_k 与对应系数 c_{kj}的线性组合.

例 1.2　外积. 矩阵-矩阵乘积的一个简单例子是外积（outer product），这是一个 m 维列向量 u 与一个 n 维行向量 v 的乘积；结果是一个秩为 1 的 $m \times n$ 矩阵. 外积可以写成

$$\left[\begin{array}{c} \\ u \\ \\ \end{array}\right] \left[\begin{array}{cccc} v_1 & v_2 & \cdots & v_n \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ v_1 u & v_2 u & \cdots & v_n u \\ & & & \end{array}\right] = \begin{bmatrix} v_1 u_1 & \cdots & v_n u_1 \\ \vdots & & \vdots \\ v_1 u_m & \cdots & v_n u_m \end{bmatrix}.$$

其列是同一个向量 u 的所有倍数，类似地，其行是同一个向量 v 的所有倍数. □

例 1.3　作为第 2 个实例，考虑 $B=AR$. 其中，R 是一个 $n \times n$ 阶上三角矩阵，其元素对$i \leqslant j$有 $r_{ij}=1$，对 $i>j$ 有 $r_{ij}=0$. 乘积可以写成

$$\left[\begin{array}{c|c|c} & & \\ b_1 & \cdots & b_n \\ & & \end{array}\right] = \left[\begin{array}{c|c|c} & & \\ a_1 & \cdots & a_n \\ & & \end{array}\right] \begin{bmatrix} 1 & \cdots & 1 \\ & \ddots & \vdots \\ & & 1 \end{bmatrix}.$$

公式（1.6）给出

$$b_j = Ar_j = \sum_{k=1}^{j} a_k. \tag{1.7}$$

即 B 的第 j 列是 A 的前 j 列的和，矩阵 R 是不定积分算子的离散模拟. □

1.4　值域和零空间

矩阵的值域（range）记为 range(A)，它是所有能表示为 Ax 形式的向量的集合. 公式（1.2）自然导出下列 range(A)的特征.

6

定理 1.1　range(A)是由 A 的列所张成的空间.

证明　由（1.2），任意的 Ax 是 A 的列的线性组合，反之，由 A 的列所张成的空间中的任意向量 y 可以写成列的一个线性组合，$y = \sum_{j=1}^{n} x_j a_j$. 由于由系数 x_j 构成向量 x，我们有 $y = Ax$，因此 y 在 A 的值域中. □

由定理 1.1，矩阵 A 的值域也称为 A 的列空间（column space）.

$A \in \mathbb{C}^{m\times n}$，其零空间（nullspace）写成 null($A$)，它是满足 $Ax = 0$ 的向量 x 的集合，这里的 0 是 $\mathbb{C}^m$ 中的 0-向量. 每个向量 $x \in$ null(A)的元素给出零作为 A 的列的线性组合展开式的系数：$0 = x_1 a_1 + x_2 a_2 + \cdots + x_n a_n$.

1.5　秩

矩阵的列秩（column rank）是它的列空间的维数，类似地，矩阵的行秩（row rank）是由其行所张成的空间的维数. 行秩总是等于列秩（在其他一些证明中，这是在第 4 讲和第 5 讲中讨论的奇异值分解的一个推论），所以我们简称这个数为矩阵的秩.

一个满秩（full rank）的 $m \times n$ 矩阵是具有最大可能的秩（m 和 n 的较小者）的矩阵. 这意味着一个$m \geqslant n$的满秩矩阵一定有 n 个线性无关的列，这样的矩阵也可以用这样的性质来表征：它所定义的映射是一一对应的.

定理 1.2　$m \geqslant n$ 的矩阵 $A \in \mathbb{C}^{m\times n}$ 为满秩的，当且仅当它不把两个不同的向量映射到同一个向量上.

证明　（⇒）如果 A 满秩，那么它的列线性无关，所以它们构成了 range(A)的一组基. 这意味着每个 $b \in$ range(A)有惟一的关于 A 的列的线性展开式，因此，由（1.2），每个 $b \in$ range(A)存在惟一的 x 使得 $b = Ax$.（⇐）反之，若 A 并非满秩，则它的列 a_j 是相关的，存在一个非平凡的线性组合使得 $\sum_{j=1}^{n} c_j a_j = 0$. 由系数 c_j 构成的非零向量 c 满足 $Ac = 0$，所以对任意 x，有 $Ax = A(x + c)$，这样 A 把不同的向量映射到同一个向量，矛盾. □

1.6 逆

非奇异（nonsingular）或可逆（invertible）的矩阵是一个满秩方形矩阵. 注意，非奇异 $m\times m$ 矩阵的 m 列构成全空间 $\mathbb{C}^m$ 的一组基，因此，我们可以把任意向量惟一地表示为它们的线性组合. 特别地，第 j 个元素为 1 而其他元素为零的标准单位向量 $\boldsymbol{e}_j$ 可以展开为：

7

$$\boldsymbol{e}_j = \sum_{i=1}^{m} z_{ij}\boldsymbol{a}_i. \tag{1.8}$$

令 $\boldsymbol{Z}$ 是元素为 z_{ij} 的矩阵，再令 z_j 记为 $\boldsymbol{Z}$ 的第 j 列，则（1.8）可以写成 $\boldsymbol{e}_j = \boldsymbol{A}z_j$. 这个方程具有（1.6）的形式. 它可以更简明地写成

$$\left[\begin{array}{c|c|c} \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_m \end{array}\right] = \boldsymbol{I} = \boldsymbol{AZ},$$

其中，$\boldsymbol{I}$ 是 $m\times m$ 的单位矩阵. 矩阵 $\boldsymbol{Z}$ 是 $\boldsymbol{A}$ 的逆（inverse）. 任意非奇异的方阵 $\boldsymbol{A}$ 有惟一的逆，记为 $\boldsymbol{A}^{-1}$，满足 $\boldsymbol{AA}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$.

下列定理收录了方阵非奇异的一些等价条件，这些条件在线性代数教科书中都会提及，这里我们不给出定理的证明. 关于（f），参看第 5 讲.

定理 1.3　对 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$，下面条件是等价的：

(a) $\boldsymbol{A}$ 有逆 $\boldsymbol{A}^{-1}$,

(b) $\mathrm{rank}(\boldsymbol{A}) = m$,

(c) $\mathrm{range}(\boldsymbol{A}) = \mathbb{C}^m$,

(d) $\mathrm{null}(\boldsymbol{A}) = \{0\}$,

(e) 0 不是 $\boldsymbol{A}$ 的特征值,

(f) 0 不是 $\boldsymbol{A}$ 的奇异值,

(g) $\det(\boldsymbol{A}) \neq 0$.

(g)中，我们提到了行列式，虽然它在理论上是一个方便的概念，但是在数值算法却难得找到用武之地.

1.7 矩阵的逆乘向量

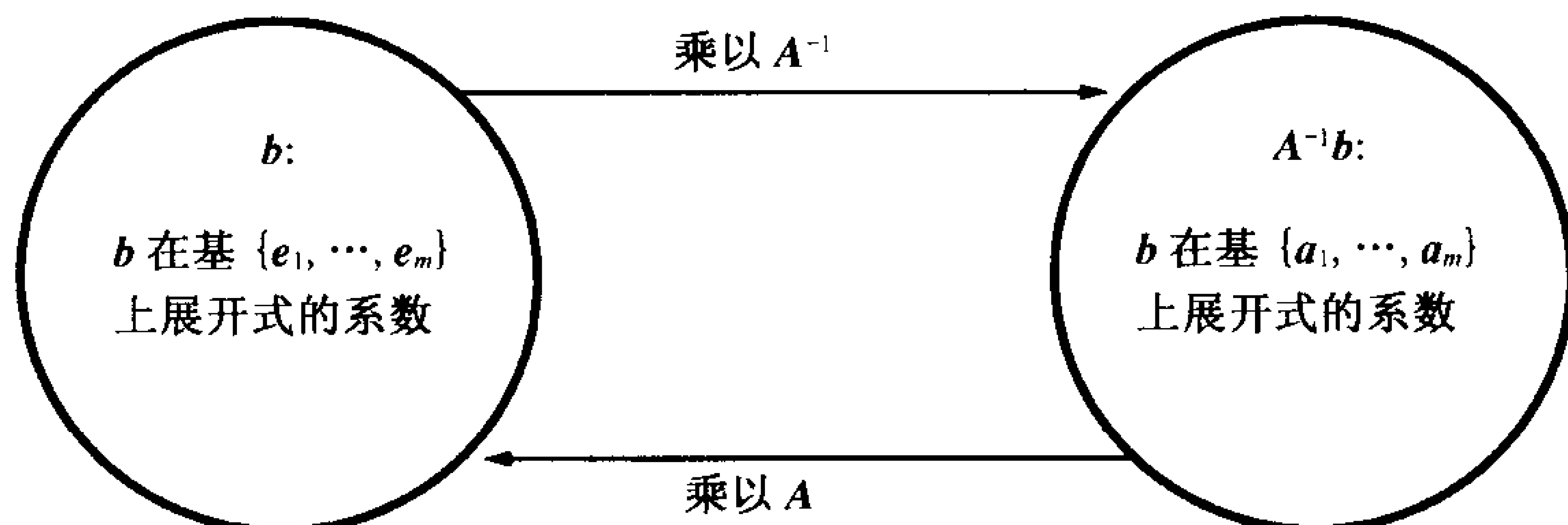
当写出乘积 $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ 时，重要的是不要让逆矩阵的记号把我们弄迷糊了！不要把 $\boldsymbol{x}$ 理解为 $\boldsymbol{A}^{-1}$ 作用到 $\boldsymbol{b}$ 上的结果，我们应该理解为它是满足方程 $\boldsymbol{Ax} = \boldsymbol{b}$ 的惟一向量. 由（1.2），这意味着 $\boldsymbol{x}$ 是 $\boldsymbol{b}$ 在 $\boldsymbol{A}$ 的列的基上的惟一线性展开式的系数所构成的

向量.

下面这点怎么强调都不过分，所以复述如下：

$A^{-1}b$ 为 b 在 A 的列的基上展开式的系数所构成的向量.

8 乘以 A^{-1} 是一个改变基（change of basis）的运算：



在这个描述中，符号"b"的含义是不同的，有时"b"记为一个 m-数组，而有时它又作为一个抽象向量空间中的一个点. 读者应该对这一点多加琢磨，直到适应这种区分为止.

1.8 关于 m 和 n 的注记

数值线性代数中习惯将矩形矩阵的维数记为 $m \times n$，本书遵循这个习惯.

如果矩阵是方阵，则如何？通常的习惯是记它的维数为 $n \times n$，但在本书中我们一般用 $m \times m$. 我们很多的算法需要考虑取方阵的列的一个子集构成的子矩形矩阵. 如果子矩阵是 $m \times n$ 的，则原矩阵最好是 $m \times m$ 的.

习　题

1.1 令 B 为一个 4×4 矩阵，对它做下列运算：

(1) 2 倍第 1 列，

(2) $\frac{1}{2}$第 3 行，

(3) 把第 3 行加到第 1 行，

(4) 交换第 1 列和第 4 列，

(5) 除第 2 行外的每行减去第 2 行，

(6) 以第 3 列代替第 4 列，

(7) 删去第 1 列（这样列维数减 1）.

(a) 把结果写成 8 个矩阵的乘积.

(b) 再写成 3 个矩阵的乘积 ABC（同样的 B）.

1.2 假设质点 m_1, m_2, m_3, m_4 放在一条直线的 x_1, x_2, x_3, x_4 位置上，它们用弹簧系数为

k_{12},k_{23},k_{34}的弹簧相连接，弹簧延伸的自然长度为 $\ell_{12},\ell_{23},\ell_{34}$. 令 f_1,f_2,f_3,f_4 记为作用在质点上向右的力，例如，$f_1=k_{12}(x_2-x_1-\ell_{12})$.

9

(a) 写出关于列向量 $\boldsymbol{f}$ 和 $\boldsymbol{x}$ 的 4×4 矩阵方程，在这个方程中，令 $\boldsymbol{K}$ 表示该矩阵.

(b) 在物理意义下，$\boldsymbol{K}$ 的元素的量纲是什么（例如是质量乘时间，还是距离除以质量，等等）?

(c) 在物理意义下，$\det(\boldsymbol{K})$的量纲是什么？

(d) 假设基于米、千克和秒的单位制下给出了 $\boldsymbol{K}$ 的数值，将此系统基于厘米、克和秒单位制用矩阵 $\boldsymbol{K}'$重写. $\boldsymbol{K}'$到 $\boldsymbol{K}$ 是什么关系？$\det(\boldsymbol{K}')$到 $\det(\boldsymbol{K})$是什么关系？

1.3　推广例 1.3，我们称一个元素为 r_{ij}的方阵或矩形矩阵 $\boldsymbol{R}$ 是上三角的（upper-triangular），如果对 $i>j$ 有 $r_{ij}=0$. 利用（1.8）考虑由 $\boldsymbol{R}$ 的前 n 列张成的空间，由此证明如果 $\boldsymbol{R}$ 是一个非奇异的 $m\times m$ 上三角矩阵，则 $\boldsymbol{R}^{-1}$ 也是上三角的.（类似的结论对下三角矩阵也成立.）

1.4　令 $f_1,\cdots,f_8$ 为一个定义在区间［1,8］上的函数集合，它具有性质：对任意的数 $d_1,\cdots,d_8$，存在系数 $c_1,\cdots,c_8$ 的集合，使得

$$\sum_{j=1}^{8} c_j f_j(i) = d_i,\quad i=1,\cdots,8 .$$

(a) 由本讲所述定理，证明 $d_1,\cdots,d_8$ 惟一确定 $c_1,\cdots,c_8$.

(b) 令 $\boldsymbol{A}$ 表示由数据 $d_1,\cdots,d_8$ 到系数 $c_1,\cdots,c_8$ 的线性映射的 8×8 矩阵，$\boldsymbol{A}^{-1}$的 i，j 元素是什么？

10

第2讲　正交向量和矩阵

自从20世纪60年代以来，很多数值线性代数最好的算法都是基于这样或那样的正交性. 本讲介绍有关的内容：正交向量和正交（酉）矩阵.

2.1　伴　　随

一个数 z 的复共轭（complex conjugate），记为 $\bar{z}$ 或 z^*，是将其虚部取负号获得的. 对于实的 z，$\bar{z}=z$.

一个 $m\times n$ 矩阵 $\boldsymbol{A}$ 的埃米尔特共轭（Hermite conjugate）或伴随（adjoint），记为 $\boldsymbol{A}^*$，其 i，j 元素为 $\boldsymbol{A}$ 的 j，i 元素的复共轭. 例如

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \Rightarrow \boldsymbol{A}^*=\begin{bmatrix} \bar{a}_{11} & \bar{a}_{21} & \bar{a}_{31} \\ \bar{a}_{12} & \bar{a}_{22} & \bar{a}_{32} \end{bmatrix}.$$

如果 $\boldsymbol{A}=\boldsymbol{A}^*$，$\boldsymbol{A}$ 为埃米尔特矩阵. 由定义，埃米尔特矩阵一定是方阵. 对于实的 $\boldsymbol{A}$，伴随只是简单地改变它的行和列. 在此情形中，伴随也称为转置（transpose），写成 $\boldsymbol{A}^{\mathrm{T}}$. 如果一个实矩阵是埃米尔特矩阵，即 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$，也称它为对称（symmetric）矩阵.

多数数值线性代数教科书假设所讨论的矩阵是实的，所以主要用^{T}代替*. 然而，[11] 我们采用另外的做法，这是因为多数论述的思想本质上并不限于实矩阵. 例如，在本书中行向量通常记作 a^* 而不 a^{T}. 有些读者即使倾向于想像所有的量都是实的且*和^{T}是同义的，也不太会遇到麻烦.

2.2　内　　积

两个列向量 $\boldsymbol{x}$，$\boldsymbol{y}\in\mathbb{C}^m$ 的内积（inner product）是 $\boldsymbol{x}$ 的伴随与 $\boldsymbol{y}$ 的乘积：

$$\boldsymbol{x}^*\boldsymbol{y}=\sum_{i=1}^{m}\bar{x}_i y_i. \tag{2.1}$$

$\boldsymbol{x}$ 的欧几里得长度可以写成 $\|\boldsymbol{x}\|$（这样的向量范数将在下一讲系统地讨论），它可以定义为 $\boldsymbol{x}$ 与自己内积的平方根：

$$\|\boldsymbol{x}\|=\sqrt{\boldsymbol{x}^*\boldsymbol{x}}=\left(\sum_{i=1}^{m}|x_i|^2\right)^{1/2}. \tag{2.2}$$

$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 之间的角的余弦也可由内积表示为

$$\cos\alpha = \frac{\boldsymbol{x}^*\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}. \tag{2.3}$$

如同这里一样，在本书的各处，我们要提到代数公式的几何解释. 对些这些几何解释，读者可以想像向量是实的而非复的，虽然通常这些解释可以用这样或那样的方式推广到在复的情形.

内积是双线性（bilinear）的，它意味着对于每个向量分别都是线性的：

$$(\boldsymbol{x}_1+\boldsymbol{x}_2)^*\boldsymbol{y} = \boldsymbol{x}_1^*\boldsymbol{y}+\boldsymbol{x}_2^*\boldsymbol{y},$$
$$\boldsymbol{x}^*(\boldsymbol{y}_1+\boldsymbol{y}_2) = \boldsymbol{x}^*\boldsymbol{y}_1+\boldsymbol{x}^*\boldsymbol{y}_2,$$
$$(\alpha\boldsymbol{x})^*(\beta\boldsymbol{y}) = \bar{\alpha}\beta\boldsymbol{x}^*\boldsymbol{y}.$$

对于任意相容维数的矩阵或向量 $\boldsymbol{A}$ 和 $\boldsymbol{B}$，我们还将频繁地使用如下容易证明的性质

$$(\boldsymbol{AB})^* = \boldsymbol{B}^*\boldsymbol{A}^*. \tag{2.4}$$

这个公式类似于同等重要的可逆方阵的乘积公式

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}. \tag{2.5}$$

记号 $\boldsymbol{A}^{-*}$ 是 $(\boldsymbol{A}^*)^{-1}$ 或 $(\boldsymbol{A}^{-1})^*$ 的简记. 这两个矩阵是相等的，可用（2.4）中取 $\boldsymbol{B}=\boldsymbol{A}^{-1}$ 来证明. 12

2.3 正交向量

一对向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 若满足 $\boldsymbol{x}^*\boldsymbol{y}=0$，它们就是正交的（orthogonal）. 若 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 都是实的，这就意味着它们在 $\mathbb{R}^m$ 相互成直角. 两个向量的集合 X 和 Y，如果每个 $\boldsymbol{x}\in X$ 正交于每个 $\boldsymbol{y}\in Y$，就称 X 和 Y 正交（也称 X 正交于 Y).

一个非零向量的集合 S，如果它的元素两两正交，也就是若对 $\boldsymbol{x},\boldsymbol{y}\in S$，$\boldsymbol{x}\neq\boldsymbol{y}\Rightarrow\boldsymbol{x}^*\boldsymbol{y}=0$，那么它是正交集合. 如果向量集合 S 是正交的，此外每个 $\boldsymbol{x}\in S$ 都有 $\|\boldsymbol{x}\|=1$，称 S 是标准正交的（orthonormal).

定理 2.1　正交集合 S 中的向量是线性无关的.

证明　如果 S 中的向量不是无关的，则存在某个 $\boldsymbol{v}_k\in S$，它可表示成其他元素 $\boldsymbol{v}_1,\cdots,\boldsymbol{v}_n\in S$ 的线性组合

$$\boldsymbol{v}_k = \sum_{\substack{i=1\\ i\neq k}}^{n} c_i\boldsymbol{v}_i.$$

因 $\boldsymbol{v}_k\neq 0$，所以 $\boldsymbol{v}_k^*\boldsymbol{v}_k=\|\boldsymbol{v}_k\|^2>0$，用内积的双线性和 S 的正交性，计算得

$$\boldsymbol{v}_k^*\boldsymbol{v}_k = \sum_{\substack{i=1\\ i\neq k}}^{n} c_i\boldsymbol{v}_k^*\boldsymbol{v}_i = 0,$$

这和 S 中的向量为非零的假设矛盾. □

作为定理2.1的一个推论，可知若一个正交集合 $S\subseteq\mathbb{C}^m$ 包含 m 个向量，则它是 $\mathbb{C}^m$ 的一组基.

2.4　向量的分量

内积和正交性的概念所描述的最重要的思想是：内积可以用于将任意向量分解为正交分量.

例如，假设$\{\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_n\}$是一个正交集合，令 $\boldsymbol{v}$ 为一个任意向量，量 $\boldsymbol{q}_j^*\boldsymbol{v}$ 是一个数量. 利用这些数量作为一个展开式的坐标，得到向量

$$\boldsymbol{r}=\boldsymbol{v}-(\boldsymbol{q}_1^*\boldsymbol{v})\boldsymbol{q}_1-(\boldsymbol{q}_2^*\boldsymbol{v})\boldsymbol{q}_2-\cdots-(\boldsymbol{q}_n^*\boldsymbol{v})\boldsymbol{q}_n \tag{2.6}$$

正交于$\{\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_n\}$. 这可以由计算 $\boldsymbol{q}_i^*\boldsymbol{r}$ 来验证：

$$\boldsymbol{q}_i^*\boldsymbol{r}=\boldsymbol{q}_i^*\boldsymbol{v}-(\boldsymbol{q}_1^*\boldsymbol{v})(\boldsymbol{q}_i^*\boldsymbol{q}_1)-\cdots-(\boldsymbol{q}_n^*\boldsymbol{v})(\boldsymbol{q}_i^*\boldsymbol{q}_n).$$

因为对 $i\neq j$ 有 $\boldsymbol{q}_i^*\boldsymbol{q}_j=0$，这个和式简化为：

13

$$\boldsymbol{q}_i^*\boldsymbol{r}=\boldsymbol{q}_i^*\boldsymbol{v}-(\boldsymbol{q}_i^*\boldsymbol{v})(\boldsymbol{q}_i^*\boldsymbol{q}_i)=0.$$

因此，我们看到 $\boldsymbol{v}$ 可以分解为 $n+1$ 个正交分量：

$$\boldsymbol{v}=\boldsymbol{r}+\sum_{i=1}^{n}(\boldsymbol{q}_i^*\boldsymbol{v})\boldsymbol{q}_i=\boldsymbol{r}+\sum_{i=1}^{n}(\boldsymbol{q}_i\boldsymbol{q}_i^*)\boldsymbol{v}. \tag{2.7}$$

在这个分解式中，$\boldsymbol{r}$ 是向量 $\boldsymbol{v}$ 正交于向量集合$\{\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_n\}$的部分，或等价地说，正交于由这个向量集合所张成的子空间，且$(\boldsymbol{q}_i^*\boldsymbol{v})\boldsymbol{q}_i$ 是 $\boldsymbol{v}$ 在 $\boldsymbol{q}_i$ 方向的部分.

如果$\{\boldsymbol{q}_i\}$是$\mathbb{C}^m$ 的一组基，则 n 一定等于 m，而且 $\boldsymbol{r}$ 一定是零向量，这样 $\boldsymbol{v}$ 完全分解成 m 个在 $\boldsymbol{q}_i$ 方向的正交分量：

$$\boldsymbol{v}=\sum_{i=1}^{m}(\boldsymbol{q}_i^*\boldsymbol{v})\boldsymbol{q}_i=\sum_{i=1}^{m}(\boldsymbol{q}_i\boldsymbol{q}_i^*)\boldsymbol{v}. \tag{2.8}$$

在（2.7）和（2.8）两式中，我们用两种不同的方法写出公式，先用$(\boldsymbol{q}_i^*\boldsymbol{v})\boldsymbol{q}_i$ 再用$(\boldsymbol{q}_i\boldsymbol{q}_i^*)\boldsymbol{v}$. 这两个表达式是相等的，但有不同的解释. 在第一种情形中，我们把 $\boldsymbol{v}$ 作为系数 $\boldsymbol{q}_i^*\boldsymbol{v}$ 乘向量 $\boldsymbol{q}_i$ 的和. 在第二种情形中，我们把 $\boldsymbol{v}$ 作为 $\boldsymbol{v}$ 在各个方向 $\boldsymbol{q}_i$ 上正交投影之和. 第 i 个投影运算是由非常特殊的秩1矩阵 $\boldsymbol{q}_i\boldsymbol{q}_i^*$ 得到的. 在第6讲中我们将讨论这个（以及其他的）投影的过程.

2.5　酉　矩　阵

如果方阵 $Q\in\mathbb{C}^{m\times m}$满足 $Q^*=Q^{-1}$，即 $Q^*Q=I$，称 Q 是酉的（unitary）在实数

情形也称为正交的（orthogonal）. 利用 Q 的列，$Q^*Q=I$ 可以写成

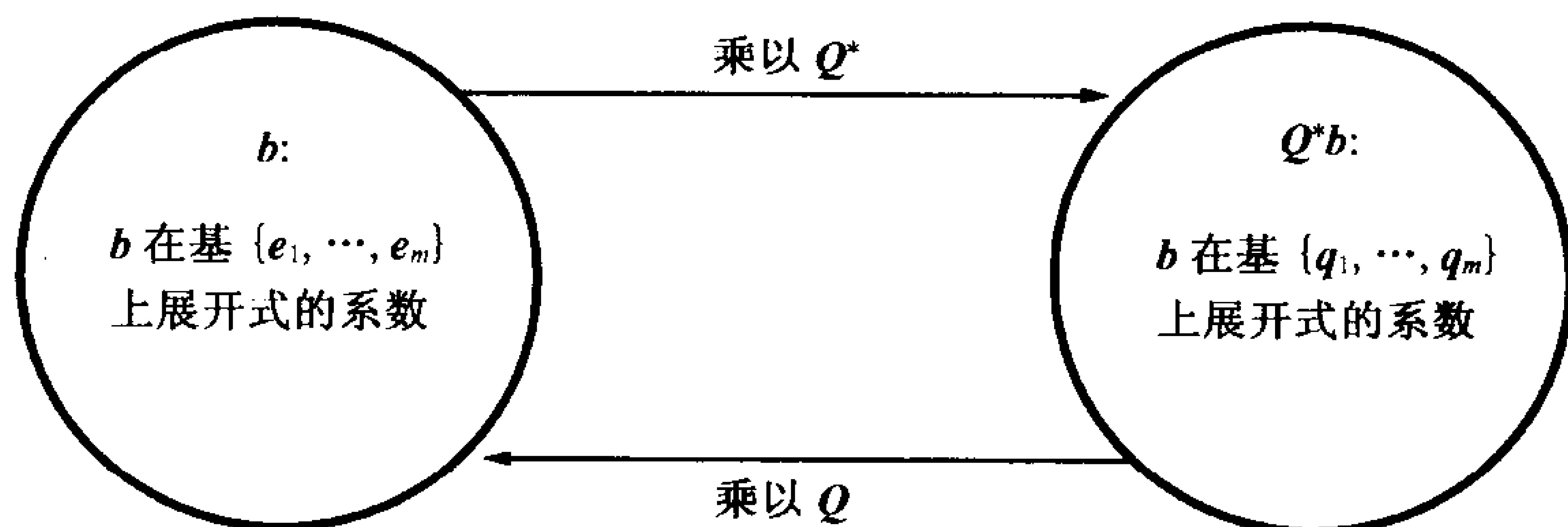
$$\begin{bmatrix} \text{——} \; q_1^* \; \text{——} \\ \text{——} \; q_2^* \; \text{——} \\ \vdots \\ \text{——} \; q_m^* \; \text{——} \end{bmatrix} \begin{bmatrix} q_1 & q_2 & \cdots & q_m \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}.$$

换句话说，$q_i^*q_j=\delta_{ij}$，且一个酉矩阵的列组成$\mathbb{C}^m$ 的一组正交基. 符号 δ_{ij}称为克罗内克 δ，当 $i\neq j$ 时，它等于0，而当 $i=j$ 时，它等于1.

2.6　乘以酉矩阵

上一讲讨论了矩阵-向量乘积 Ax 和 $A^{-1}b$ 的解释，如果 A 是酉矩阵 Q，这些乘积就是 Qx 和 Q^*b，同样的解释当然仍有效. 如前所述，Qx 是 Q 的列的线性组合，系数为 x. 反之，Q^*b 是 b 在 Q 的列的基上展开式的系数所构成的向量. [14]

这可以图解为：



乘以一个酉矩阵或其伴随矩阵的这些过程保持了欧几里得意义下的几何结构，因为内积是保持不变的. 即对酉矩阵 Q，容易由（2.4）验证

$$(Qx)^*(Qy)=x^*y, \tag{2.9}$$

内积的不变性意味着向量之间的角度保持不变，这对它们的长度也成立：

$$\|Qx\|=\|x\|. \tag{2.10}$$

在实数情形下，乘以正交矩阵 Q 对应于向量空间的一个刚性旋转（当 $\det Q=1$ 时）或镜射（当 $\det Q=-1$ 时）.

习　　题

2.1　证明若矩阵 A 同时是三角矩阵和酉矩阵时，则它是对角矩阵.

2.2　毕达哥拉斯定理指出，对 n 个正交向量的集合 $\{x_i\}$，有

$$\left\| \sum_{i=1}^{n} x_i \right\|^2 = \sum_{i=1}^{n} \|x_i\|^2.$$

(a) 在 $n=2$ 的情形，通过直接计算 $\|x_1+x_2\|^2$ 证明此结论.

(b) 用归纳法证明以上计算式对一般情形也成立.

2.3 令 $A\in\mathbb{C}^{m\times m}$ 为埃米尔特矩阵，A 的特征向量是使 $Ax=\lambda x$ 成立的非零向量 $x\in\mathbb{C}^m$，其中 $\lambda\in\mathbb{C}$ 是对应的特征值.

15

(a) 证明 A 的所有特征值都是实数.

(b) 证明若 x 和 y 是对应不同特征值的特征向量，则 x 和 y 正交.

2.4 酉矩阵的特征值有什么性质?

2.5 令 $S\in\mathbb{C}^{m\times m}$ 是反埃米尔特矩阵，即 $S^*=-S$.

(a) 用习题 2.3 证明 S 的特征值是纯虚数.

(b) 证明 $I-S$ 非奇异.

(c) 矩阵 $Q=(I-S)^{-1}(I+S)$ 称为 S 的凯莱变换（Cayley transform)，证明它是酉矩阵.（这是一个类似线性分式变换 $(1+s)/(1-s)$ 的矩阵，该变换将复平面 s 的左半部分保形地映射到单位圆.）

2.6 若 u 和 v 是 m 维向量，矩阵 $A=I+uv^*$ 称为单位矩阵的一个秩 1 扰动（rank-one perturbation of the identity). 证明若 A 非奇异，则它的逆可写成 $A^{-1}=I+\alpha uv^*$，对数量 α 成立，并给出 α 的表示式，对于什么样的 u 和 v，A 是奇异的? 若 A 奇异，null(A)是什么?

2.7 阿达马矩阵（Hadamard matrix）是所有元素均为 ± 1 的矩阵，且它的转置等于它的逆乘上一个常数因子. 众所周知，若 A 是一个维数为 $m>2$ 的阿达马矩阵，则 m 是 4 的倍数. 然而不知道对每个这样的 m，是否都存在一个阿达马矩阵，虽然已经找到了 $m\leqslant 424$ 所有情形的例子. 证明下面的递推关系

$$H_0=[1], \qquad H_{k+1}=\begin{bmatrix} H_k & H_k \\ H_k & -H_k \end{bmatrix}.$$

16

给出了每个维数为 $m=2^k,(k=0,1,2,\cdots)$ 的阿达马矩阵.

第3讲 范　　数

范数集中描述了向量空间中大小和距离这些重要概念，它是整个数值线性代数中衡量逼近和收敛性的尺度.

3.1 向量范数

范数（norm）是一个函数$\|\cdot\|$: $\mathbb{C}^m \to \mathbb{R}$，它给定每个向量一个实值的长度. 为了符合通常的长度意义，范数必须满足以下三个条件，即对于所有的向量$\boldsymbol{x}$和$\boldsymbol{y}$与所有的数$\alpha \in \mathbb{C}$，

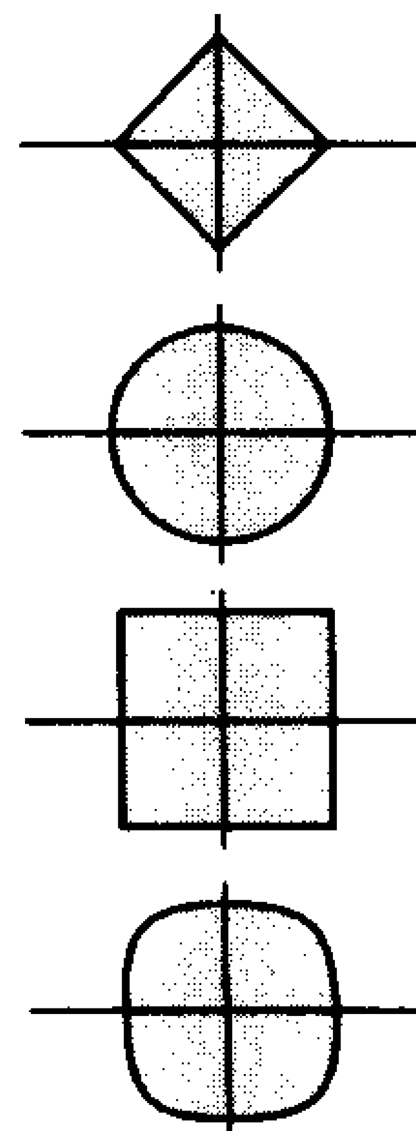
$$\begin{aligned} &(1)\quad \|\boldsymbol{x}\| \geqslant 0, \text{且只当} \boldsymbol{x}=0 \text{时} \|\boldsymbol{x}\| = 0, \\ &(2)\quad \|\boldsymbol{x}+\boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|, \\ &(3)\quad \|\alpha \boldsymbol{x}\| = |\alpha| \|\boldsymbol{x}\|. \end{aligned} \tag{3.1}$$

从字面上看，这些条件要求如下：（1）非零向量的范数是正数，（2）向量和的范数不超过它各向量范数之和——即三角不等式（triangle inequality），（3）与一个向量成比例的向量，其范数也以同样的比例常数成比例.

在上一讲中，我们用$\|\cdot\|$表示为欧几里得长度函数（向量各元素平方和的平方根）. 然而，(3.1) 的三个条件允许长度有不同的概念，具有这种灵活性常常是很有用的.

p-范数是最重要的一类向量范数，定义如下. 其中$m=2$时对应于每一个范数的闭单位球$\{\boldsymbol{x} \in \mathbb{C}^m : \|\boldsymbol{x}\| \leqslant 1\}$，图示在右侧.

$$\begin{aligned} \|\boldsymbol{x}\|_1 &= \sum_{i=1}^{m} |x_i|, \\ \|\boldsymbol{x}\|_2 &= \left(\sum_{i=1}^{m} |x_i|^2 \right)^{1/2} = \sqrt{\boldsymbol{x}^* \boldsymbol{x}}, \\ \|\boldsymbol{x}\|_\infty &= \max_{1 \leqslant i \leqslant m} |x_i|, \\ \|\boldsymbol{x}\|_p &= \left(\sum_{i=1}^{m} |x_i|^p \right)^{1/p} (1 \leqslant p < \infty). \end{aligned} \tag{3.2}$$



2-范数是欧几里得长度函数，它的单位球是圆球. 航空公司用1-范数来确定手提箱的最大容许尺寸. 瑞典斯德哥尔摩的 Sergel 广场呈现了4-范数单位球的形状. 丹麦诗

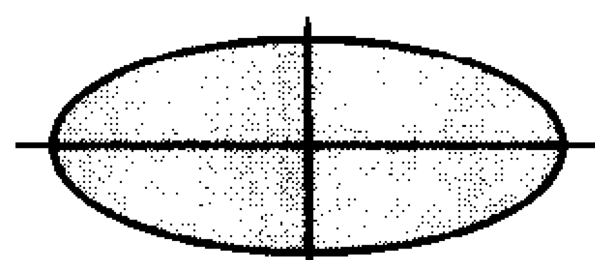
人 Piet Hein 使这个“超椭圆”成为一种广泛流行的悦目外形，例如很多会议桌就常采用这种形状.

除了 p-范数外，最有用的范数是*加权p-范数*（weighted p-norm），其中每个向量空间的坐标都给出了自身的权. 一般对于任意的范数 $\|\cdot\|$，加权范数可以写成

$$\|\boldsymbol{x}\|_W = \|\boldsymbol{W}\boldsymbol{x}\|. \tag{3.3}$$

这里 $\boldsymbol{W}$ 是一个对角矩阵，其第 i 个对角元素是权 $w_i \neq 0$. 例如，$\mathbb{C}^m$ 上的权 2-范数 $\|\cdot\|_W$ 规定如下：

$$\|\boldsymbol{x}\|_W = \left(\sum_{i=1}^{m} |w_i x_i|^2\right)^{1/2}. \tag{3.4}$$



也可以把权范数的思想推广到容许 $\boldsymbol{W}$ 是任何一个非奇异矩阵，不必非要是对角的（习题 3.1）.

在本书中最重要的范数是无权的 2-范数以及它导出的矩阵范数.

3.2　由向量范数导出的矩阵范数

一个 $m\times n$ 矩阵可以看成 mn 维空间的向量：矩阵的 mn 个元素的每一个都是独立的坐标. 因此任意的 mn 维范数都可用于度量这个矩阵的“大小”.

18

然而，讨论矩阵空间时，某些特殊的范数比起在（3.2）~（3.3）所讨论的向量范数更加有用. 这就是*导出矩阵范数*（induced matrix norm），它把矩阵作为它的标准定义域和值域之间的算子，根据算子的行为来定义范数.

在 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$ 的定义域和值域上分别给出向量范数 $\|\cdot\|_{(n)}$ 和 $\|\cdot\|_{(m)}$，导出矩阵范数 $\|\boldsymbol{A}\|_{(m,n)}$ 是使不等式

$$\|\boldsymbol{A}\boldsymbol{x}\|_{(m)} \leqslant C\|\boldsymbol{x}\|_{(n)}. \tag{3.5}$$

对所有 $\boldsymbol{x}\in\mathbb{C}^n$ 成立的最小的数 C. 换句话说，$\|\boldsymbol{A}\|_{(m,n)}$ 是取遍所有向量 $\boldsymbol{x}\in\mathbb{C}^n$，比值 $\|\boldsymbol{A}\boldsymbol{x}\|_{(m)}/\|\boldsymbol{x}\|_{(n)}$ 的上确界——$\boldsymbol{A}$ 可以“拉伸”向量 $\boldsymbol{x}$ 的最大因子. 我们称 $\|\cdot\|_{(m,n)}$ 为由 $\|\cdot\|_{(m)}$ 和 $\|\cdot\|_{(n)}$ 导出的矩阵范数.

由（3.1）的条件（3）知 $\boldsymbol{A}$ 的作用由它对单位向量的作用所决定. 因而，矩阵范数可以等价地由单位向量在 $\boldsymbol{A}$ 作用下的像来定义：

$$\|\boldsymbol{A}\|_{(m,n)} = \sup_{\substack{\boldsymbol{x}\in\mathbb{C}^n\\ \boldsymbol{x}\neq 0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_{(m)}}{\|\boldsymbol{x}\|_{(n)}} = \sup_{\substack{\boldsymbol{x}\in\mathbb{C}^n\\ \|\boldsymbol{x}\|_{(n)}=1}} \|\boldsymbol{A}\boldsymbol{x}\|_{(m)}. \tag{3.6}$$

这个形式的定义，如同上面（3.2）的图那样，很方便地直观表示了导出矩阵范数.

3.3 几个例子

例 3.1 矩阵

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} \tag{3.7}$$

映射$\mathbb{C}^2$到$\mathbb{C}^2$. 它也映射$\mathbb{R}^2$到$\mathbb{R}^2$，因为A的系数是实的，如果我们要画出图形就更方便了，同时也可以完全确定矩阵的p-范数.

图3-1描绘了A对$\mathbb{R}^2$中单位球的作用，单位球由1-，2-，以及∞-范数所定义，由该图可看出A的3种范数的图形解释. 与范数无关，A把$e_1=(1,0)^*$映射到A的第1列，即e_1自身，把$e_2=(0,1)^*$映射到A的第2列，即$(2,2)^*$. 对1-范数，被A放大最大的单位向量x是$(0,1)^*$（或其负向量），其放大因子是4. 对∞-范数，被A放大最大的单位向量x是$(1,1)^*$（或其负向量），放大因子为3. 对2-范数，被A放大最大的单位向量是图中虚线所示的向量（或其负向量），放大因子为2.9208.（注意，它至少是$\sqrt{8}\approx 2.8284$，因为$(0,1)^*$映射到$(2,2)^*$.）在第5讲将要考虑如何计算这样的2-范数. □ 19

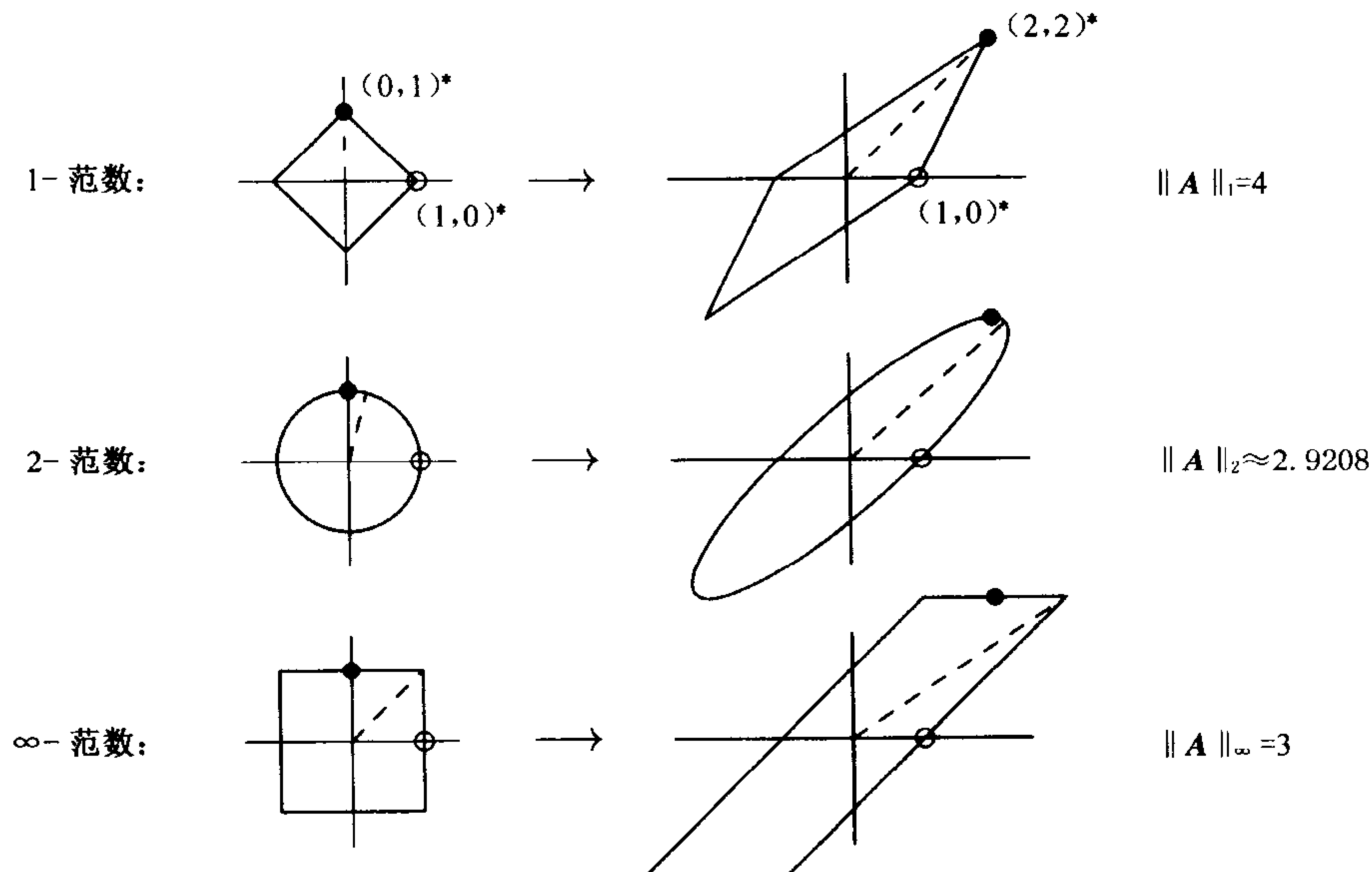


图3-1 左边是$\mathbb{R}^2$对于$\|\cdot\|_1$，$\|\cdot\|_2$和$\|\cdot\|_\infty$的单位球，右边是它们在(3.7)的矩阵A下的像，虚线表示标出各种范数情形下，由A放大最大的向量

例 3.2 对角矩阵的 p-范数. 令 D 为对角矩阵

$$D=\begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_m \end{bmatrix}.$$

则如图 3-1 的第 2 行，2-范数单位球在 D 下的像是一个 m 维椭圆，它的半轴长由数 $|d_i|$ 给出. 由 D 放大最大的单位向量是那些映射到椭圆最长半轴（长度为 $\max_i\{|d_i|\}$）上的向量. 因此有 $\|D\|_2=\max_{1\leqslant i\leqslant m}\{|d_i|\}$，在下一讲将看到每个矩阵把 2-范数单位球映射到一个椭圆——若 $m>2$，严格地说应称为超椭圆（hyperellipse）——尽管其轴可任意取向.

对 2-范数的这个结论可以推广到任意的 p：若 D 是对角矩阵，则 $\|D\|_p=\max_{1\leqslant i\leqslant m}|d_i|$.

□

20 **例 3.3 矩阵的 1-范数.** 若 A 为任意 $m\times n$ 矩阵，则 $\|A\|_1$ 等于 A 的"最大列和". 我们给出解释和推导如下，按列写出 A

$$A=\begin{bmatrix} \Bigg| & & \Bigg| \\ a_1 & \cdots & a_n \\ \Bigg| & & \Bigg| \end{bmatrix}, \tag{3.8}$$

其中，每个 a_j 是 m 维向量. 考虑（3.2）所示的菱形 1-范数单位球，即集合 $\left\{x\in\mathbb{C}^n:\sum_{j=1}^{n}|x_j|\leqslant1\right\}$. 这个集合的像中的任意向量 Ax 满足：

$$\|Ax\|_1=\left\|\sum_{j=1}^{n}x_ja_j\right\|_1\leqslant\sum_{j=1}^{n}|x_j|\,\|a_j\|_1\leqslant\max_{1\leqslant j\leqslant n}\|a_j\|_1.$$

因此，导出矩阵 1-范数满足 $\|A\|_1\leqslant\max_{1\leqslant j\leqslant n}\|a_j\|_1$. 选取 $x=e_j$，其中，j 使 $\|a_j\|_1$ 最大，即可达到此界，因此矩阵范数为

$$\|A\|_1=\max_{1\leqslant j\leqslant n}\|a_j\|_1. \tag{3.9}$$

□

例 3.4 矩阵的 ∞-范数. 同理，可以证明 $m\times n$ 矩阵的 ∞-范数等于"最大行和".

$$\|A\|_\infty=\max_{1\leqslant i\leqslant m}\|a_i^*\|_1, \tag{3.10}$$

其中，a_i^* 记为 A 的第 i 行. □

3.4 柯西-施瓦茨不等式和赫尔德不等式

$p\neq 1, \infty$时，计算矩阵的p-范数更为困难，为研究这个问题，注意内积可以用p-范数确定其界. 令p和q满足$1/p+1/q=1$，其中$1\leqslant p, q\leqslant\infty$. 赫尔德不等式(Hölder inequality)指出，对任意向量$\boldsymbol{x}$和$\boldsymbol{y}$，

$$|\boldsymbol{x}^*\boldsymbol{y}| \leqslant \|\boldsymbol{x}\|_p\|\boldsymbol{y}\|_q. \tag{3.11}$$

柯西-施瓦茨不等式(Cauchy-Schwarz inequality)是其$p=q=2$的特殊情形：

$$|\boldsymbol{x}^*\boldsymbol{y}| \leqslant \|\boldsymbol{x}\|_2\|\boldsymbol{y}\|_2. \tag{3.12}$$

这个结论的推导可以在线性代数教科书中找到. 这两个式子的界都是紧密的，意即对某些选择的$\boldsymbol{x}$和$\boldsymbol{y}$，不等式会成为等式.

例3.5 行向量的2-范数. 考虑只含一行的矩阵，写成$\boldsymbol{A}=\boldsymbol{a}^*$，其中$\boldsymbol{a}$是一个列向量. 可由柯西-施瓦茨不等式得到导出矩阵2-范数. 对任意$\boldsymbol{x}$，有$\|\boldsymbol{A}\boldsymbol{x}\|_2=|\boldsymbol{a}^*\boldsymbol{x}|\leqslant\|\boldsymbol{a}\|_2\|\boldsymbol{x}\|_2$. 观察$\|\boldsymbol{A}\boldsymbol{a}\|_2=\|\boldsymbol{a}\|_2^2$，可知不等式的界是紧密的. 因此有

$$\|\boldsymbol{A}\|_2=\sup_{x\neq 0}\{\|\boldsymbol{A}\boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2\}=\|\boldsymbol{a}\|_2. \qquad \square$$

21

例3.6 外积的2-范数. 更一般地，考虑秩1外积$\boldsymbol{A}=\boldsymbol{u}\boldsymbol{v}^*$，其中，$\boldsymbol{u}$是$m$维向量而$\boldsymbol{v}$是$n$维向量. 对任意的$n$维向量$\boldsymbol{x}$，可如下定$\|\boldsymbol{A}\boldsymbol{x}\|_2$的界：

$$\|\boldsymbol{A}\boldsymbol{x}\|_2=\|\boldsymbol{u}\boldsymbol{v}^*\boldsymbol{x}\|_2=\|\boldsymbol{u}\|_2|\boldsymbol{v}^*\boldsymbol{x}|\leqslant\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2\|\boldsymbol{x}\|_2. \tag{3.13}$$

因此，$\|\boldsymbol{A}\|_2\leqslant\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2$. 当$x=v$时，这个不等式是个等式，即$\|\boldsymbol{A}\|_2=\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2$. $\square$

3.5 导出矩阵范数$\|AB\|$的界

矩阵乘积的导出矩阵范数也可确定其界. 令$\|\cdot\|_{(l)}$，$\|\cdot\|_{(m)}$和$\|\cdot\|_{(n)}$分别为在$\mathbb{C}^l$，$\mathbb{C}^m$和$\mathbb{C}^n$的范数，令$\boldsymbol{A}$为$l\times m$矩阵，$\boldsymbol{B}$为$m\times n$矩阵. 对任意的$x\in\mathbb{C}^n$，有

$$\|\boldsymbol{A}\boldsymbol{B}\boldsymbol{x}\|_{(l)}\leqslant\|\boldsymbol{A}\|_{(l,m)}\|\boldsymbol{B}\boldsymbol{x}\|_{(m)}\leqslant\|\boldsymbol{A}\|_{(l,m)}\|\boldsymbol{B}\|_{(m,n)}\|\boldsymbol{x}\|_{(n)}.$$

因此，$\boldsymbol{AB}$的导出范数一定满足

$$\|\boldsymbol{A}\boldsymbol{B}\|_{(l,n)}\leqslant\|\boldsymbol{A}\|_{(l,m)}\|\boldsymbol{B}\|_{(m,n)}. \tag{3.14}$$

一般地，该不等式等号不成立。例如，不等式$\|\boldsymbol{A}^n\|\leqslant\|\boldsymbol{A}\|^n$对任意方阵由向量范数导出的矩阵范数成立，但一般对$n\geqslant 2$，$\|\boldsymbol{A}^n\|=\|\boldsymbol{A}\|^n$并不成立.

3.6　一般矩阵范数

如上所述，矩阵范数不一定由向量范数导出. 一般地，矩阵范数只需满足应用在矩阵的 mn 维向量空间上的三个向量范数条件（3.1）：

$$\begin{aligned}&(1)\ \|A\|\geqslant 0,\text{且}\|A\|=0\text{ 仅当 }A=0,\\&(2)\ \|A+B\|\leqslant\|A\|+\|B\|,\\&(3)\ \|\alpha A\|=|\alpha|\|A\|.\end{aligned}\tag{3.15}$$

最重要的矩阵范数是希尔伯特-施密特范数（Hilbert-Schmidt norm）或弗罗贝尼乌斯范数（Frobenius norm），但它不是由向量范数导出的，它定义为

$$\|A\|_F=\left(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\right)^{1/2}.\tag{3.16}$$

22 可以看到，这是把矩阵看成 mn 维向量时的 2-范数. 弗罗贝尼乌斯范数的公式可根据各行或各列来写. 例如，若 $\boldsymbol{a}_j$ 是 A 的第 j 列，则有

$$\|A\|_F=\left(\sum_{j=1}^{n}\|\boldsymbol{a}_j\|_2^2\right)^{1/2}.\tag{3.17}$$

这个等式，还有基于用行代替列类似的结果可以简洁地表示为方程

$$\|A\|_F=\sqrt{\operatorname{tr}(A^*A)}=\sqrt{\operatorname{tr}(AA^*)},\tag{3.18}$$

其中，$\operatorname{tr}(B)$ 记为 B 的迹（trace），即其对角元素之和.

如同导出矩阵范数，弗罗贝尼乌斯范数也可以用于确定矩阵乘积的界. 令 $C=AB$，其元素为 c_{ik}，令 $\boldsymbol{a}_i^*$ 记为 A 的第 i 行，$\boldsymbol{b}_j$ 记为 B 的第 j 列. 则有 $c_{ij}=\boldsymbol{a}_i^*\boldsymbol{b}_j$，由柯西-施瓦茨不等式得 $|c_{ij}|\leqslant\|\boldsymbol{a}_i\|_2\|\boldsymbol{b}_j\|_2$. 两边平方，并对所有 i，j 求和，得到

$$\begin{aligned}\|AB\|_F^2&=\sum_{i=1}^{n}\sum_{j=1}^{m}|c_{ij}|^2\\&\leqslant\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\|\boldsymbol{a}_i\|_2\|\boldsymbol{b}_j\|_2\right)^2\\&=\sum_{i=1}^{n}(\|\boldsymbol{a}_i\|_2)^2\sum_{j=1}^{m}\left(\|\boldsymbol{b}_j\|_2\right)^2=\|A\|_F^{\ 2}\|B\|_F^2.\end{aligned}$$

3.7　酉乘积下的不变性

类似于向量的 2-范数，矩阵 2-范数特殊性质之一是乘以酉矩阵的运算下的不变性，同样的性质对弗罗贝尼乌斯范数也成立.

定理3.1 对任意的$A\in\mathbb{C}^{m\times n}$和酉矩阵$Q\in\mathbb{C}^{m\times m}$，有

$$\|QA\|_2=\|A\|_2,\quad \|QA\|_F=\|A\|_F.$$

证明 因为由（2.10），$\|Qx\|_2=\|x\|_2$对每个x成立，由（3.6）可得2-范数的不变性，对弗罗贝尼乌斯范数要用到（3.18）. □

如果把Q推广到具有正交列的矩形矩阵（即$Q\in\mathbb{C}^{p\times m}$，$p>m$），定理3.1仍然成立. 对于右乘酉矩阵，或更一般右乘具有正交行的矩形矩阵，类似的等式也成立. 23

习　题

3.1 证明若W是任意的非奇异矩阵，则由（3.3）定义的函数$\|\cdot\|_W$是一种向量范数.

3.2 令$\|\cdot\|$记为$\mathbb{C}^m$上的任意范数，也记为其在$\mathbb{C}^{m\times m}$上的导出矩阵范数. 证明$\rho(A)\leqslant\|A\|$，其中，$\rho(A)$是A的谱半径（spectral radius），即A的特征值λ的最大绝对值$|\lambda|$.

3.3 向量和矩阵的p-范数通过各种不等式相联系，它们常常涉及维数m或n，证明下列每个不等式，并给出一个非零向量或矩阵（对一般m,n）使不等式取等号的例子. 本题中，x是一个m维向量，A是$m\times n$矩阵.

(a) $\|x\|_\infty\leqslant\|x\|_2$，

(b) $\|x\|_2\leqslant\sqrt{m}\|x\|_\infty$，

(c) $\|A\|_\infty\leqslant\sqrt{n}\|A\|_2$，

(d) $\|A\|_2\leqslant\sqrt{m}\|A\|_\infty$.

3.4 令A为$m\times n$矩阵，B是A的一个$\mu\times v(\mu\leqslant m,v\leqslant n)$子矩阵，它是由选出$A$的某些行和某些列得到的矩阵.

(a) 如同习题1.1第7步那样，解释怎样以某些行和列的“删节矩阵”乘A得到B.

(b) 用此乘积，证明$\|B\|_p\leqslant\|A\|_p$对任意满足$1\leqslant p\leqslant\infty$的$p$成立.

3.5 例3.6表明，若E是外积$E=uv^*$，则$\|E\|_2=\|u\|_2\|v\|_2$. 同样的性质对弗罗贝尼乌斯范数（即$\|E\|_F=\|u\|_F\|v\|_F$）也成立吗？证明这个事实或给出一个反例.

3.6 令$\|\cdot\|$记为$\mathbb{C}^m$的任一范数，它的对偶范数（dual norm）$\|\cdot\|'$由公式$\|x\|'=\sup\limits_{\|y\|=1}|y^*x|$定义.

(a) 证明$\|\cdot\|'$是一种范数.

(b) 给出$x,y\in\mathbb{C}^m$，满足$\|x\|=\|y\|=1$. 证明存在一个秩1矩阵$B=yz^*$，使得$Bx=y$，且$\|B\|=1$，其中，$\|B\|$是B的由向量范数$\|\cdot\|$导出的矩阵范数. 可以不加证明地用下述引理：给定$x\in\mathbb{C}^m$，则存在非零的$z\in\mathbb{C}^m$，使得$|z^*x|=\|z\|'\|x\|$. 24

第4讲 奇异值分解

奇异值分解（singular value decomposition，SVD）是一种矩阵因子分解，这一计算在很多算法中都要用到. SVD在分析概念时也是同等重要的. 在很多线性代数问题中，如果我们首先思考若做SVD，情况将会怎样，那么问题可能会得到更好的理解.

4.1 几何的观测

SVD受到下面几何事实的启发：

单位球面在任意 $m\times n$ 矩阵下的像是一个超椭圆.

SVD可以应用于实矩阵和复矩阵. 然而，在描述几何解释时，通常假设矩阵是实的.

"超椭圆"一词可能不为人熟知，它正是椭圆的 m 维推广. 我们可以将$\mathbb{R}^m$中的超椭圆定义为在某些正交方向 $\boldsymbol{u}_1,\cdots,\boldsymbol{u}_m\in\mathbb{R}^m$，以某些因子 $\sigma_1,\cdots,\sigma_m$（可能为零）延伸$\mathbb{R}^m$中的单位球面所得到的曲面. 习惯上取 $\boldsymbol{u}_i$ 为单位向量，即 $\|\boldsymbol{u}_i\|_2=1$. 向量$\{\sigma_i\boldsymbol{u}_i\}$是超椭圆的主半轴（principal semiaxe)，其长度为 $\sigma_1,\cdots,\sigma_m$. 若 $\boldsymbol{A}$ 有秩 r，则恰有 r 个长度 $\boldsymbol{\sigma}_i$ 为非零；特别地，若 $m\geqslant n$，则最多有 n 个为非零.

25

单位球面的像有下面的含义：单位球面是指 n 维空间中通常的Euclid球面，即2-范数意义下的单位球面，记之为 S. 而在映射 A 下 S 的像 AS 就是所定义的超椭圆.

这个几何事实并非显然. 我们将用线性代数的语言重新叙述它，并在稍后加以证明. 目前先假设它是正确的.

令 S 为$\mathbb{R}^n$中的单位球面，取任意的 $A\in\mathbb{R}^{m\times n}$，其中 $m\geqslant n$. 为简单起见，现暂设 $\boldsymbol{A}$ 有满秩 n. 像 $\boldsymbol{AS}$ 是$\mathbb{R}^m$中一个超椭圆，现根据 $\boldsymbol{AS}$ 的形状确定 $\boldsymbol{A}$ 的某些性质. 关键的概念如图4-1所示.

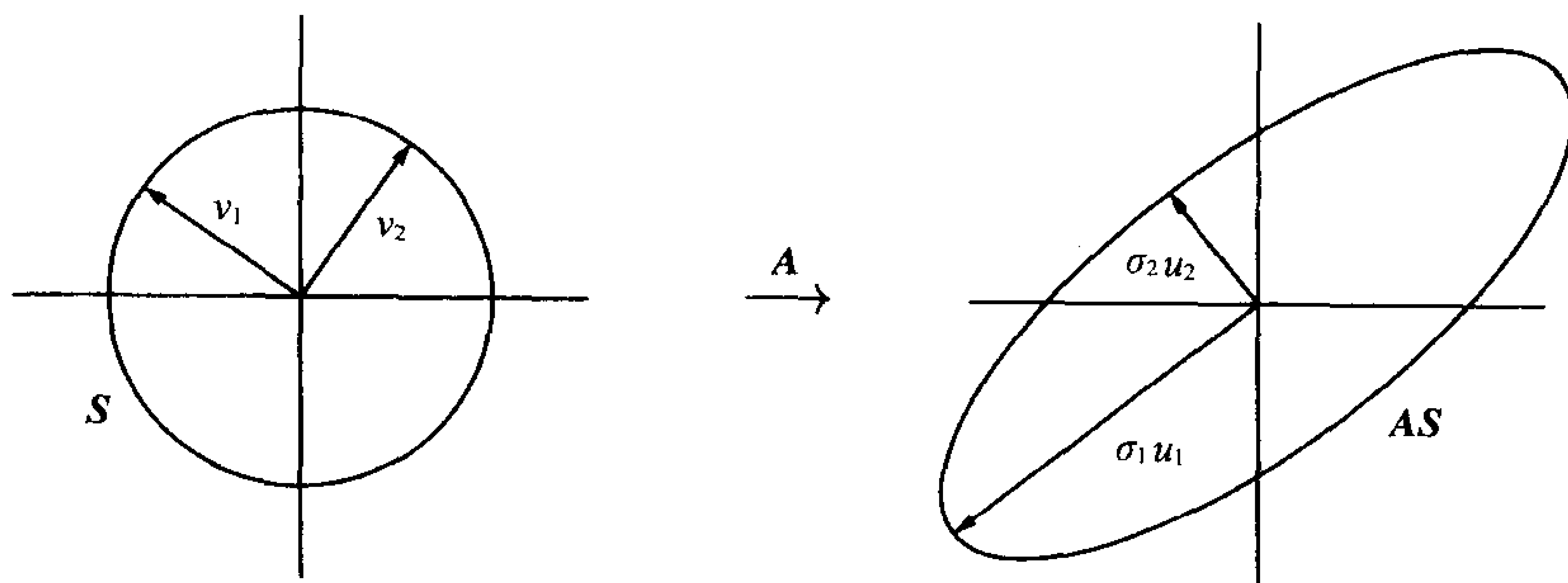


图4-1　2×2矩阵的SVD

首先，我们定义A的n个奇异值（singular value）．它们是AS的n个主半轴的长度，写成$\sigma_1,\sigma_2,\cdots,\sigma_n$，习惯上假设奇异值以降序编号，$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n > 0$.

其次，我们定义A的n个左奇异向量（left singular vector）．它们是取在AS主半轴方向的单位向量$\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n\}$，其编号对应于奇异值．向量$\sigma_i \mu_i$就是$AS$的第$i$大的主半轴.

最后，我们定义A的n个右奇异向量（right singular vector）．它们是单位向量$\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n\} \in S$，是$AS$主半轴的原像，编号使得$Av_j = \sigma_j u_j$.

上述定义中的“左”和“右”显然是难于处理的，它们来自下面（4.2）和（4.3）式中因子$\boldsymbol{U}$和$\boldsymbol{V}$的位置．困难是在图 4-1 中，左奇异向量对应图中右边的部分，而右奇异向量对应左边的部分！可以这样解决此问题，即交换图中的两半，并且映射A由右指向左，但这将会和固有的习惯背道而驰.

26

4.2　约　化　SVD

上面已指出联系右奇异向量$\{\boldsymbol{v}_j\}$和左奇异向量$\{\boldsymbol{u}_j\}$的方程可以写成

$$A\boldsymbol{v}_j = \sigma_j \boldsymbol{u}_j, \qquad 1 \leqslant j \leqslant n. \tag{4.1}$$

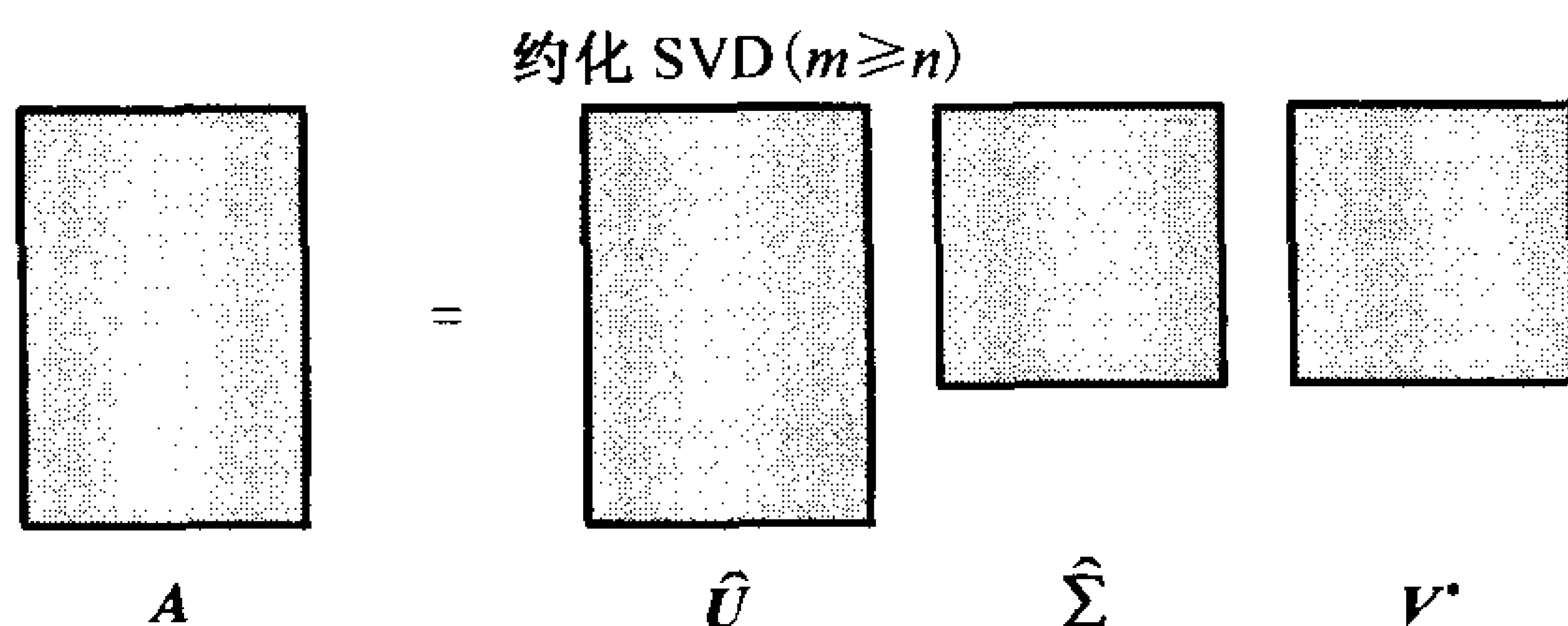
这个向量方程组可以表示为一个矩阵方程

$$\begin{bmatrix} \\ A \\ \\ \end{bmatrix}\left[\begin{array}{c|c|c|c} & & & \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_n \\ & & & \end{array}\right]\begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix},$$

或更紧凑地写成$\boldsymbol{AV} = \hat{U}\hat{\Sigma}$．在此矩阵方程中，$\hat{\Sigma}$是一个$n \times n$矩阵，它有正的实元素（因为假设$A$有满秩$n$），$\hat{U}$是一个有正交列的$m \times n$矩阵．$\boldsymbol{V}$是一个具有正交列的$n \times n$矩阵，因此$\boldsymbol{V}$是酉的，可以右乘其逆$\boldsymbol{V}^*$，得到

$$A = \hat{U}\hat{\Sigma}V^*. \tag{4.2}$$

这个A的因子分解称为A的约化奇异值分解（reduced singular value decomposition）或约化 SVD（reduced SVD）．图解为



4.3 完 全 SVD

在大多数的应用中，SVD 正好有刚才描述的形式．然而，这并不是教科书中通常表述 SVD 思路的方法．为了区分分解式（4.2）与更标准的“完全”SVD，我们引用了“约化”一词，并在 U 和 Σ 上带上帽子（记号^）．“约化”与“完全”的术语和帽子记号将保持在全书中，而且也有类似的约化与完全 QR 因子分解之分．这些惯例的提示印在本书的文前部分．

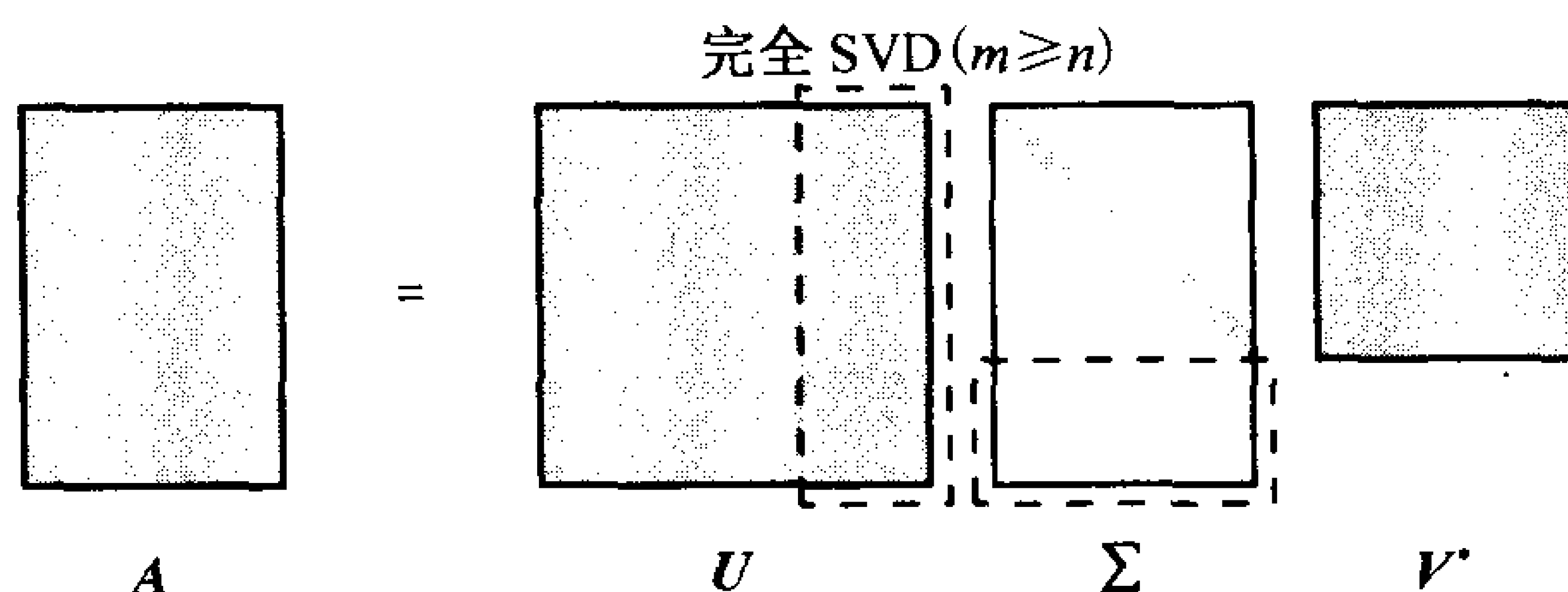
27

概念叙述如下，$\hat{U}$的列是 m 维空间$\mathbb{C}^m$ 中 n 个正交向量，除非 $m=n$，它们不能组成$\mathbb{C}^m$ 的一组基，而且$\hat{U}$也不是一个酉矩阵．然而，添上附加的 $m-n$ 个正交向量，$\hat{U}$可扩展为一个酉矩阵．我们可以以任意的方式这样做，其结果称为 $\boldsymbol{U}$.

若在（4.2）中$\hat{\boldsymbol{U}}$以 $\boldsymbol{U}$ 代替，则 $\hat{\Sigma}$ 也要改变．为了使乘积保持不变，$\boldsymbol{U}$ 的最后 $m-n$列可乘以零．这样 Σ 是一个 $m\times n$ 矩阵，它在上面 $n\times n$ 的块上为 $\hat{\Sigma}$，而下面的 $m-n$ 行为零．现在有了一个新的因子分解，即 A 的完全 *SVD*（full SVD）：

$$A = U\Sigma V^*. \tag{4.3}$$

这里，$\boldsymbol{U}$ 是 $m\times m$ 酉矩阵，$\boldsymbol{V}$ 是 $n\times n$ 酉矩阵，而 Σ 是 $m\times n$ 的，其对角是正的实元素．图解为：



图中虚线指出了 $\boldsymbol{U}$ 中“不起作用”的列和 Σ 中“不起作用”的行，它们是由（4.3）到（4.2）所删去的部分.

叙述了完全 SVD，现在可以放弃 $\boldsymbol{A}$ 是满秩的这样一个为了简化所做的假设．如果 $\boldsymbol{A}$ 是秩亏损的，因子分解（4.3）仍成立．所有的变化只是由超椭圆的几何性质所决定的 $\boldsymbol{A}$ 的左奇异向量，它现在不是 n 个而只是 r 个．为了构造酉矩阵 $\boldsymbol{U}$，我们引入 $m-r$ 列附加的任意正交列而不是刚才的 $m-n$ 列．矩阵 $\boldsymbol{V}$ 也需要 $n-r$ 列任意的正交列来扩展由几何性质决定的 r 列．矩阵 Σ 现在要有 r 个正的对角元素，而余下的 $n-r$ 个元素等于零.

同理，约化 SVD（4.2）对于非满秩的矩阵 $\boldsymbol{A}$ 也有意义，可取$\hat{\boldsymbol{U}}$为 $m\times n$ 的，$\hat{\Sigma}$ 的维数为 $n\times n$ 的，其某些对角元素为零，或进一步压缩其表达式，使$\hat{\boldsymbol{U}}$为 $m\times r$ 的并

且$\hat{\Sigma}$是$r\times r$的，其对角元素是严格正的.

4.4　正式的定义

令m和n为任意的数，并不要求$m\geqslant n$，给定$A\in\mathbb{C}^{m\times n}$，不必要是满秩的，$A$的一个奇异值分解是这样的因子分解 28

$$A=U\Sigma V^{*} \tag{4.4}$$

其中

$$U\in\mathbb{C}^{m\times m}\text{是酉矩阵,}$$
$$V\in\mathbb{C}^{n\times n}\text{是酉矩阵,}$$
$$\Sigma\in\mathbb{R}^{m\times n}\text{是对角矩阵.}$$

此外，假设Σ的对角元素σ_j非负且以非增的次序排列，即$\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_p\geqslant 0$，其中$p=\min(m,n)$.

注意，即使A不是方阵，对角矩阵Σ与A有相同的形状，但U和V总是正方形的酉矩阵.

显然，$\mathbb{R}^n$中单位球面在映射$A=U\Sigma V^{*}$下的像一定是$\mathbb{R}^m$中一个超椭圆. 酉映射V^{*}保持了球面不变，对角矩阵Σ将球面拉伸到一个有标准基的超椭圆，最后酉映射U旋转或镜射这个超椭圆，但不改变它的形状. 因此，如果我们能证明每个矩阵有一个 SVD，我们就证明了单位球面在任意线性映射下的像是一个超椭圆，正如本讲开头所说那样.

4.5　存在性和惟一性

定理 4.1　每个矩阵$A\in\mathbb{C}^{m\times n}$有一个奇异值分解 (4.4). 此外，奇异值$\{\sigma_j\}$是惟一确定的，而且若$A$是方阵以及$\sigma_j$是相异的，则左、右奇异向量$\{u_j\}$和$\{v_j\}$惟一确定到复符号因子 (即绝对值为 1 的复数量因子).

证明　为证明 SVD 的存在性，我们将A中有最大作用的方向分隔出来，然后在A的维数上使用归纳法.

令$\sigma_1=\|A\|_2$，由范数的紧性，一定存在向量$v_1\in\mathbb{C}^n$和$u_1\in\mathbb{C}^m$，满足$\|v_1\|_2=\|u_1\|_2=1$及$Av_1=\sigma_1 u_1$. 将v_1任意扩张为$\mathbb{C}^n$的一组正交基$\{v_j\}$，u_1任意扩张为$\mathbb{C}^m$的一组正交基，令U_1和V_1分别记为以u_j和v_j为列的酉矩阵. 则有

$$U_1^{*}AV_1=S=\begin{bmatrix}\sigma_1 & w^{*}\\ 0 & B\end{bmatrix}, \tag{4.5}$$

其中，**0** 是 $m-1$ 维的列向量，w^* 是 $n-1$ 维的行向量，而 B 有维数 $(m-1)\times(n-1)$. 此外，

29

$$\left\|\begin{bmatrix}\sigma_1 & w^*\\ 0 & B\end{bmatrix}\begin{bmatrix}\sigma_1\\ w\end{bmatrix}\right\|_2 \geq \sigma_1^2+w^*w=(\sigma_1^2+w^*w)^{1/2}\left\|\begin{bmatrix}\sigma_1\\ w\end{bmatrix}\right\|_2,$$

这隐含 $\|S\|_2\geq(\sigma_1^2+w^*w)^{1/2}$. 因为 U_1 和 V_1 都是酉阵，可知 $\|S\|_2=\|A\|_2=\sigma_1$，这隐含 $w=0$.

如果 $n=1$ 或 $m=1$，我们已经证明了存在性，另外，子矩阵 B 描述了 A 在正交于 v_1 的子空间上的作用. 由归纳法假设，B 有 SVD$B=U_2\Sigma_2V_2^*$. 容易验证

$$A=U_1\begin{bmatrix}1 & 0\\ 0 & U_2\end{bmatrix}\begin{bmatrix}\sigma_1 & 0\\ 0 & \Sigma_2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & V_2\end{bmatrix}^* V_1{}^*\Rightarrow\begin{bmatrix}3 & 0\\ 0 & -2\end{bmatrix}$$

是 A 的一个 SVD，这就完成了存在性的证明.

对于惟一性的结论，几何的验证是直接的：如果超椭圆的半轴长是相异的，则半轴自身除符号外，其余由几何性质决定了. 代数的证明如下，首先，记 $\sigma_1=\|A\|_2$，它是由（4.4）所得到的这个条件惟一确定的. 现在对 v_1 假设有另一个线性无关的向量 w 满足 $\|w\|_2=1$ 及 $\|Aw\|_2=\sigma_1$，定义正交于 v_1 的单位向量 v_2，它是 v_1 和 w 的线性组合

$$v_2=\frac{w-(v_1{}^*w)v_1}{\|w-(v_1{}^*w)v_1\|_2}.$$

因为 $\|A\|_2=\sigma_1$，所以 $\|Av_2\|_2\leq\sigma_1$. 但这一定是个等式，否则，因对某两个常数 c 和 s，有 $w=v_1c+v_2s$，且 $|c|^2+|s|^2=1$，将会有 $\|Aw\|_2<\sigma_1$. 此向量 v_2 是 A 对应于奇异值 σ_1 的第 2 个右奇异向量. 它将导致向量 y（等于 $V_1{}^*v_2$ 的最后 $n-1$ 个分量）的出现，y 满足 $\|y\|_2=1$ 及 $\|By\|_2=\sigma_1$，最后可断言，若奇异向量 v_1 不惟一，则对应的奇异值 σ_1 不是单一的. 为完成惟一性的证明，我们指出，如上所述，一旦 σ_1，v_1 及 u_1 确定，SVD 余下的部分由 A 在与 v_1 正交的空间上的作用确定. 因 v_1 除符号外是确定的，则这个正交的空间惟一确定，由归纳法，可以得到余下的奇异值和奇异向量的惟一性. □

习　题

4.1　求下列矩阵的 SVD（手算）：

30

$$\text{(a)}\begin{bmatrix}3 & 0\\ 0 & -2\end{bmatrix},\ \text{(b)}\begin{bmatrix}2 & 0\\ 0 & 3\end{bmatrix},\ \text{(c)}\begin{bmatrix}0 & 2\\ 0 & 0\\ 0 & 0\end{bmatrix},\ \text{(d)}\begin{bmatrix}1 & 1\\ 0 & 0\end{bmatrix},\ \text{(e)}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}.$$

4.2　设A是$m\times n$矩阵，B是在纸面上将A顺时针旋转90°所得的$n\times m$矩阵（这不是严格的标准数学变换!）. A和B是否有相同的奇异值? 证明答案是肯定的，或给出一个反例.

4.3　写出一个MATLAB程序（参看第9讲），功能如下：给出一个实的2×2矩阵A，如图4-1那样在单位圆上画出右奇异向量v_1和v_2，在相应的椭圆上画出左奇异向量. 将程序应用到矩阵（3.7）和习题4.1的2×2矩阵上.

4.4　两个矩阵$A, B\in\mathbb{C}^{m\times m}$，如果对某酉矩阵$Q\in\mathbb{C}^{m\times m}$，有$A=QBQ^*$，称$A$，$B$是酉等价的（unitarily equivalent）. A和B酉等价当且仅当它们有共同的奇异值，这个命题是真的还是假的?

4.5　定理4.1指出，每个$A\in\mathbb{C}^{m\times n}$有一个SVD $A=U\Sigma V^*$. 证明若A是实的，则它有一个实的SVD（$U\in\mathbb{R}^{m\times m}$，$V\in\mathbb{R}^{n\times n}$）. 31

第 5 讲　SVD 的进一步讨论

我们继续奇异值分解的讨论，这里强调的是它与矩阵在 2-范数和弗罗贝尼乌斯范数意义下低秩逼近的联系.

5.1　基的改变

SVD 使我们可以说，如果对定义域和值域空间使用适当的基的话，则每个矩阵都是对角的.

这里来说明基的改变将会带来什么变化. 任意的 $\boldsymbol{b}\in\mathbb{C}^m$ 可在 $\boldsymbol{A}$ 的左奇异向量（$\boldsymbol{U}$ 的列）组成的基中展开，任意的 $\boldsymbol{x}\in\mathbb{C}^n$ 可在 $\boldsymbol{A}$ 的右奇异向量（$\boldsymbol{V}$ 的列）组成的基中展开. 这两个展开式的坐标向量是

$$\boldsymbol{b}' = \boldsymbol{U}^*\boldsymbol{b}, \qquad \boldsymbol{x}' = \boldsymbol{V}^*\boldsymbol{x}.$$

由（4.3），关系式 $\boldsymbol{b}=\boldsymbol{A}\boldsymbol{x}$ 可由 b' 和 x' 表示为：

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} \Leftrightarrow \boldsymbol{U}^*\boldsymbol{b} = \boldsymbol{U}^*\boldsymbol{A}\boldsymbol{x} = \boldsymbol{U}^*\boldsymbol{U}\Sigma\boldsymbol{V}^*\boldsymbol{x} \Leftrightarrow \boldsymbol{b}' = \Sigma\boldsymbol{x}'.$$

一旦 $\boldsymbol{b}=\boldsymbol{A}\boldsymbol{x}$，就有 $\boldsymbol{b}'=\Sigma\boldsymbol{x}'$. 因此，当值域在 $\boldsymbol{U}$ 的列的基中表示，定义域在 $\boldsymbol{V}$ 的列的基中表示时，$\boldsymbol{A}$ 约化为对角矩阵 Σ.

5.2　SVD 与特征值分解

用一个新基将矩阵对角化的课题也出现在特征值的研究中. 一个非亏损的方阵 $\boldsymbol{A}$，如果其值域和定义域在特征向量组成的基中表示，则 $\boldsymbol{A}$ 可以表示为特征值的对角矩阵 $\boldsymbol{\Lambda}$.

如果一个矩阵 $\boldsymbol{X}\in\mathbb{C}^{m\times m}$ 的列包含了 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$ 线性无关的特征向量，则 $\boldsymbol{A}$ 的特征值分解（eigenvalue decomposition）为

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}, \tag{5.1}$$

其中，$\boldsymbol{\Lambda}$ 是 $m\times m$ 对角矩阵，其元素为 $\boldsymbol{A}$ 的特征值. 这隐含着对 $\boldsymbol{b}$，$\boldsymbol{x}\in\mathbb{C}^m$ 满足 $\boldsymbol{b}=\boldsymbol{A}\boldsymbol{x}$，如果定义

$$\boldsymbol{b}' = \boldsymbol{X}^{-1}\boldsymbol{b}, \quad \boldsymbol{x}' = \boldsymbol{X}^{-1}\boldsymbol{x},$$

则新引入的向量 $\boldsymbol{b}'$ 和 $\boldsymbol{x}'$ 满足 $\boldsymbol{b}'=\boldsymbol{\Lambda}\boldsymbol{x}'$. 特征值将在第 24 讲中系统地讨论.

SVD 和特征值分解有根本的区别. 其一是 SVD 用两组不同的基（左和右奇异向

量的集合)，而特征值分解只用一组（特征向量）；其二是 SVD 使用正交的基，而特征值分解所用的基一般不是正交的；第三，并非所有矩阵（甚至是方阵）有特征值分解，但所有的矩阵（甚至是长方形的）都有奇异值分解，如同定理 4.1 所述．在应用方面，特征值趋于侧重包含 A 的迭代形式方面的问题，诸如矩阵的幂 A^k 或指数 e^{tA}，而奇异向量趋于侧重包含 A 自身或其逆的问题.

5.3　由 SVD 得到的矩阵性质

当我们开始列出 SVD 与其他线性代数基础课题的联系时，SVD 的作用就显而易见了．在下面的定理中，假设 A 维数为 $m\times n$．令 p 为 m 和 n 的最小者，令 $r\leqslant p$ 表示 A 非零奇异值的数目，又令 $\langle x,y,\cdots,z\rangle$ 表示由向量 $x,y,\cdots,z$ 张成的空间.

定理 5.1　*A 的秩等于其非零奇异值的数目 r.*

证明　对角矩阵的秩等于它非零元素的数目，且在分解式 $A=U\Sigma V^*$ 中，U 和 V 是满秩的，所以有 $\mathrm{rank}(A)=\mathrm{rank}(\Sigma)=r$. □

定理 5.2　$\mathrm{range}(A)=\langle u_1,\cdots,u_r\rangle$，且 $\mathrm{null}(A)=\langle v_{r+1},\cdots,v_n\rangle$.

证明　这是 $\mathrm{range}(\Sigma)=\langle e_1,\cdots,e_r\rangle\subseteq\mathbb{C}^m$ 和 $\mathrm{null}(\Sigma)=\langle e_{r+1},\cdots,e_n\rangle\subseteq\mathbb{C}^n$ 的一个推论. □　[33]

定理 5.3　$\|A\|_2=\sigma_1$，$\|A\|_F=\sqrt{\sigma_1^2+\sigma_2^2+\cdots+\sigma_r^2}$.

证明　第一个结论已在定理 4.1 的证明中建立：因 $A=U\Sigma V^*$，其中 U 和 V 是酉矩阵，由定理 3.1，$\|A\|_2=\|\Sigma\|_2=\max\{|\sigma_j|\}=\sigma_1$．对第二个结论，注意到定理 3.1 以及弗罗贝尼乌斯范数在酉乘法下的不变性，得 $\|A\|_F=\|\Sigma\|_F$，以及（3.16）就给出所要证明的公式. □

定理 5.4　*A 的非零奇异值是 A^*A 和 AA^* 的非零特征值．（A^*A 和 AA^* 有相同的非零特征值.）*

证明　由计算

$$A^*A=(U\Sigma V^*)^*(U\Sigma V^*)=V\Sigma^*U^*U\Sigma V^*=V(\Sigma^*\Sigma)V^*,$$

可见 A^*A 与 $\Sigma^*\Sigma$ 相似，从而它们有相同的 n 个特征值（见第 24 讲）．对角矩阵 $\Sigma^*\Sigma$ 的特征值是 $\sigma_1^2,\sigma_2^2,\cdots,\sigma_p^2$，如果 $n>p$，还要外加 $n-p$ 个零特征值．类似的计算可用到 AA^* 的 m 个特征值上. □

定理 5.5　*若 $A=A^*$，则 A 的奇异值是 A 的特征值的绝对值.*

证明　众所周知（见习题 2.3），埃米尔特矩阵有一个正交特征向量的完备集合，而且所有特征值都是实数．等价的陈述是，(5.1) 式成立，其中 X 等于某个酉矩阵 Q，而 Λ 是实对角矩阵．这样可写成

$$A=Q\Lambda Q^*=Q|\Lambda|\mathrm{sign}(\Lambda)Q^*,\tag{5.2}$$

其中，$|\Lambda|$和 $\mathrm{sign}(\Lambda)$分别是其元素为数$|\lambda_j|$及 $\mathrm{sign}(\lambda_j)$的对角矩阵. （也可以等价地将因子 $\mathrm{sign}(\Lambda)$放在$|\Lambda|$的左边而不是右边. ）因当 $\boldsymbol{Q}$ 是酉矩阵时 $\mathrm{sign}(\Lambda)\boldsymbol{Q}^*$ 是酉的，所以（5.2）是 $\boldsymbol{A}$ 的一个 SVD，其奇异值等于$|\Lambda|$的对角元素$|\lambda_j|$. 如果希望这些数排成不增的次序，可以在（5.2）左边的酉矩阵 $\boldsymbol{Q}$ 和右边的酉矩阵 $\mathrm{sign}(\Lambda)\boldsymbol{Q}^*$ 中插入适当的置换矩阵作为因子. □

34

定理 5.6 对 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$，有 $|\det(\boldsymbol{A})| = \prod_{i=1}^{m}\sigma_i$.

证明 方阵之积的行列式是其因子的行列式之积. 进一步来说，酉矩阵的行列式的绝对值总是 1；这可由公式 $\boldsymbol{U}^*\boldsymbol{U}=\boldsymbol{I}$ 及性质 $\det(\boldsymbol{U}^*)=(\det(\boldsymbol{U}))^*$ 得到. 所以

$$|\det(\boldsymbol{A})| = |\det(\boldsymbol{U}\Sigma\boldsymbol{V}^*)| = |\det(\boldsymbol{U})||\det(\Sigma)||\det(\boldsymbol{V}^*)| = |\det(\Sigma)| = \prod_{i=1}^{m}\sigma_i.$$

□

5.4 低秩逼近

什么是 SVD？做出解释的另一途径是，考虑矩阵 $\boldsymbol{A}$ 怎样表示为低秩矩阵的和.

定理 5.7 $\boldsymbol{A}$ 是 r 个秩 1 矩阵之和：

$$\boldsymbol{A} = \sum_{j=1}^{r}\sigma_j\boldsymbol{u}_j\boldsymbol{v}_j^*. \tag{5.3}$$

证明 如果将Σ写成 r 个矩阵Σ_j之和，其中$\Sigma_j=\mathrm{diag}(0,\cdots,0,\sigma_j,0,\cdots,0)$，则由（4.3）可得（5.3）. □

有很多方法将 $m\times n$ 矩阵 $\boldsymbol{A}$ 表示为秩 1 矩阵之和，例如，$\boldsymbol{A}$ 可以写成它的 m 行之和，或它的 n 列之和，或它的 mn 个元素之和. 另一个例子是，高斯消去法约化 $\boldsymbol{A}$ 为全秩 1 矩阵之和，一个头一行和头一列为零的秩 1 矩阵，一个头两行和头两列为零的秩 1 矩阵，如此等等.

然而，公式（5.3）表示为秩 1 矩阵的分解式有更深层的性质：第 ν 个部分和收集了 $\boldsymbol{A}$ 尽可能多的能量. 这个说法中的“能量”由 2-范数或弗罗贝尼乌斯范数所定义. 我们可以用矩阵 $\boldsymbol{A}$ 以低秩矩阵的最佳逼近问题来精确地说明它.

定理 5.8 对任意满足 $0\leqslant\nu\leqslant r$ 的 ν，定义

$$\boldsymbol{A}_\nu = \sum_{j=1}^{\nu}\sigma_j\boldsymbol{u}_j\boldsymbol{v}_j^*; \tag{5.4}$$

如果 $\nu=p=\min\{m,n\}$，定义 $\sigma_{\nu+1}=0$，则

35

$$\|\boldsymbol{A}-\boldsymbol{A}_\nu\|_2 = \inf_{\substack{\boldsymbol{B}\in\mathbb{C}^{m\times n}\\ \mathrm{rank}(\boldsymbol{B})\leqslant\nu}}\|\boldsymbol{A}-\boldsymbol{B}\|_2 = \sigma_{\nu+1}.$$

证明 设存在矩阵 $\boldsymbol{B}$ 满足 $\mathrm{rank}(\boldsymbol{B})\leqslant\nu$ 使得 $\|\boldsymbol{A}-\boldsymbol{B}\|_2 < \|\boldsymbol{A}-\boldsymbol{A}_\nu\|_2=\sigma_{\nu+1}$，则存在一个

$(n-v)$ 维的子空间 $W \subseteq \mathbb{C}^n$ 使得 $w \in W \Rightarrow Bw = 0$. 由此, 对任意 $w \in W$, 有 $Aw = (A-B)w$ 及

$$\|Aw\|_2 = \|(A-B)w\|_2 \leq \|A-B\|_2 \|w\|_2 < \sigma_{v+1} \|w\|_2.$$

因此, W 是一个 $(n-v)$ 维子空间, 其中 $\|Aw\| < \sigma_{v+1}\|w\|$. 但是存在一个 $(v+1)$ 维子空间, 其中 $\|Aw\| \geq \sigma_{v+1}\|w\|$, 它即是由 A 的前 $v+1$ 个右奇异向量张成的空间. 因为这两个空间的维数之和超过了 n, 一定存在一个非零向量在这两个空间之中, 这就导出了矛盾. □

定理5.8有一个几何解释. 什么是用直线段近似超椭圆的最佳逼近? 取此线段为最长的轴. 什么是用二维椭圆作的最佳逼近? 取由最长和次长的轴张成的椭圆. 以这种方式继续下去, 每一步增加超椭圆尚未被包括的最长轴来加强这个逼近式, r 步之后, 我们得到 A 的全部. 这种思想也出现在诸如图像压缩 (见习题9.3) 和泛函分析等不同的领域中.

对于弗罗贝尼乌斯范数, 我们不加证明地叙述类似的结果.

定理5.9 对任意满足 $0 \leq v \leq r$ 的 v, (5.4) 的 A_v 也满足

$$\|A-A_v\|_F = \inf_{\substack{B \in \mathbb{C}^{m\times n} \\ \operatorname{rank}(B) \leq v}} \|A-B\|_F = \sqrt{\sigma_{v+1}^2 + \cdots + \sigma_r^2}.$$

5.5 SVD 的计算

在本讲和上一讲中, 我们考察了 SVD 的性质, 但没有考虑怎样计算它, SVD 的计算是吸引人们注意的课题. 最好的方法是用于计算特征值的算法的变种, 这将在第31讲中讨论.

一旦能够计算, SVD 就可以作为一个工具用于所有的问题. 事实上, 本讲多数定理都有计算方面的推论. 确定矩阵秩的最佳方法是计算大于一个适当选择的允许值的奇异值的总数 (定理5.1). 寻找值域或零空间的基, 最准确的方法是借助定理5.2. (若同时考虑这两个例子, QR 因子分解提供了另外一种算法, 它比较快但不总是同样准确.) 定理5.3给出了计算 $\|A\|_2$ 的标准方法, 定理5.8和5.9则是对于 $\|\cdot\|_2$ 和 $\|\cdot\|_F$ 计算低秩逼近的标准方法. 除了这些例子之外, SVD 也是最小二乘拟合、子空间的交、正则化及很多其他问题的稳健算法中的一个组成部分. 36

习 题

5.1 在例3.1中考虑了矩阵 (3.7), 连同其他的讨论得出它的2-范数近似于2.9208. 利用 SVD 求出此矩阵 $\sigma_{\min}(A)$ 和 $\sigma_{\max}(A)$ 的准确值 (在纸面上做).

5.2 利用 SVD, 证明 $\mathbb{C}^{m\times n}$ 中的任意矩阵是一个满秩矩阵序列的极限. 换句话说, 证明满秩

矩阵的集合是$\mathbb{C}^{m\times n}$中的一个稠密子集. 用2-范数来证明. （具体是哪种范数无关紧要，因为在一个有限维空间中所有范数都是等价的.）

5.3　考虑矩阵

$$A=\begin{bmatrix}-2 & 11\\ -10 & 5\end{bmatrix}.$$

(a) 在纸面上求形为$A=U\Sigma V^{\mathrm{T}}$的实SVD. SVD不是惟一的，找出一个在U和V中负号最少的SVD.

(b) 列出A的奇异值，左奇异向量和右奇异向量. 画出$\mathbb{R}^2$中单位球及其在A下的像的精细且有标注的图，将奇异向量也画出，并标注其顶点坐标.

(c) A的1-,2-,∞-和弗罗贝尼乌斯范数等于什么?

(d) 借助SVD求A^{-1}，不直接求.

(e) 求A的特征值λ_1，λ_2.

(f) 验证$\det A=\lambda_1\lambda_2$及$|\det A|=\sigma_1\sigma_2$.

(g) A把$\mathbb{R}^2$的单位球映射到一个椭圆，椭圆的面积等于什么?

5.4　设$A\in\mathbb{C}^{m\times m}$有一个SVD　$A=U\Sigma V^*$，求$2m\times 2m$埃米尔特矩阵

$$\begin{bmatrix}0 & A^*\\ A & 0\end{bmatrix}.$$

的特征值分解（5.1）.

37

第Ⅱ部分

QR 因子分解和最小二乘

- 第 6 讲　投影算子
- 第 7 讲　QR 因子分解
- 第 8 讲　格拉姆-施密特正交化
- 第 9 讲　MATLAB
- 第 10 讲　豪斯霍尔德三角形化
- 第 11 讲　最小二乘问题

第6讲 投影算子

现在进入本书的第Ⅱ部分，其主题是正交性. 我们以投影矩阵或投影算子的基础工具开始，它包括正交的和非正交的两种情形.

6.1 投影算子

投影算子（projector）是一个方阵 $\boldsymbol{P}$，满足

$$\boldsymbol{P}^2 = \boldsymbol{P}. \tag{6.1}$$

[这样的矩阵也称为幂等的(idempotent).]这个定义包括即将讨论的正交投影算子以及非正交的投影算子两种投影算子. 为避免混淆，在非正交情形可以用斜投影算子(oblique projector）这个术语.

投影算子这个词可以想像成由这样的概念所产生的，即若将一光线恰好从正方向照在子空间 range($\boldsymbol{P}$)上，则 $\boldsymbol{P}\boldsymbol{v}$ 是向量 $\boldsymbol{v}$ 投影所产生的影子，我们在此物理直观情景下继续讨论.

显然，若 $\boldsymbol{v}\in$range($\boldsymbol{P}$)，则它确实位于自己的影子上，对 $\boldsymbol{v}$ 作用投影算子的结果就是它本身.

从数学意义上说，对某个 $\boldsymbol{x}$，有 $\boldsymbol{v}=\boldsymbol{P}\boldsymbol{x}$，且

$$\boldsymbol{P}\boldsymbol{v} = \boldsymbol{P}^2\boldsymbol{x} = \boldsymbol{P}\boldsymbol{x} = \boldsymbol{v}.$$

当 $\boldsymbol{v}\neq\boldsymbol{P}\boldsymbol{v}$ 时，光线从什么方向照射呢? 一般地说，答案依赖于 $\boldsymbol{v}$，但对任何特殊的 $\boldsymbol{v}$，容易画出由 $\boldsymbol{v}$ 到 $\boldsymbol{P}\boldsymbol{v}$ 的直线：$\boldsymbol{P}\boldsymbol{v}-\boldsymbol{v}$（图6-1)，把投影算子应用到这个向量，给出结果为零：

$$\boldsymbol{P}(\boldsymbol{P}\boldsymbol{v}-\boldsymbol{v}) = \boldsymbol{P}^2\boldsymbol{v}-\boldsymbol{P}\boldsymbol{v} = 0.$$

这意味着 $\boldsymbol{P}\boldsymbol{v}-\boldsymbol{v}\in$null($\boldsymbol{P}$). 即光线的方向对不同的 $\boldsymbol{v}$ 可能是不同的，但它总是用 null($\boldsymbol{P}$)中的一个向量来描述.

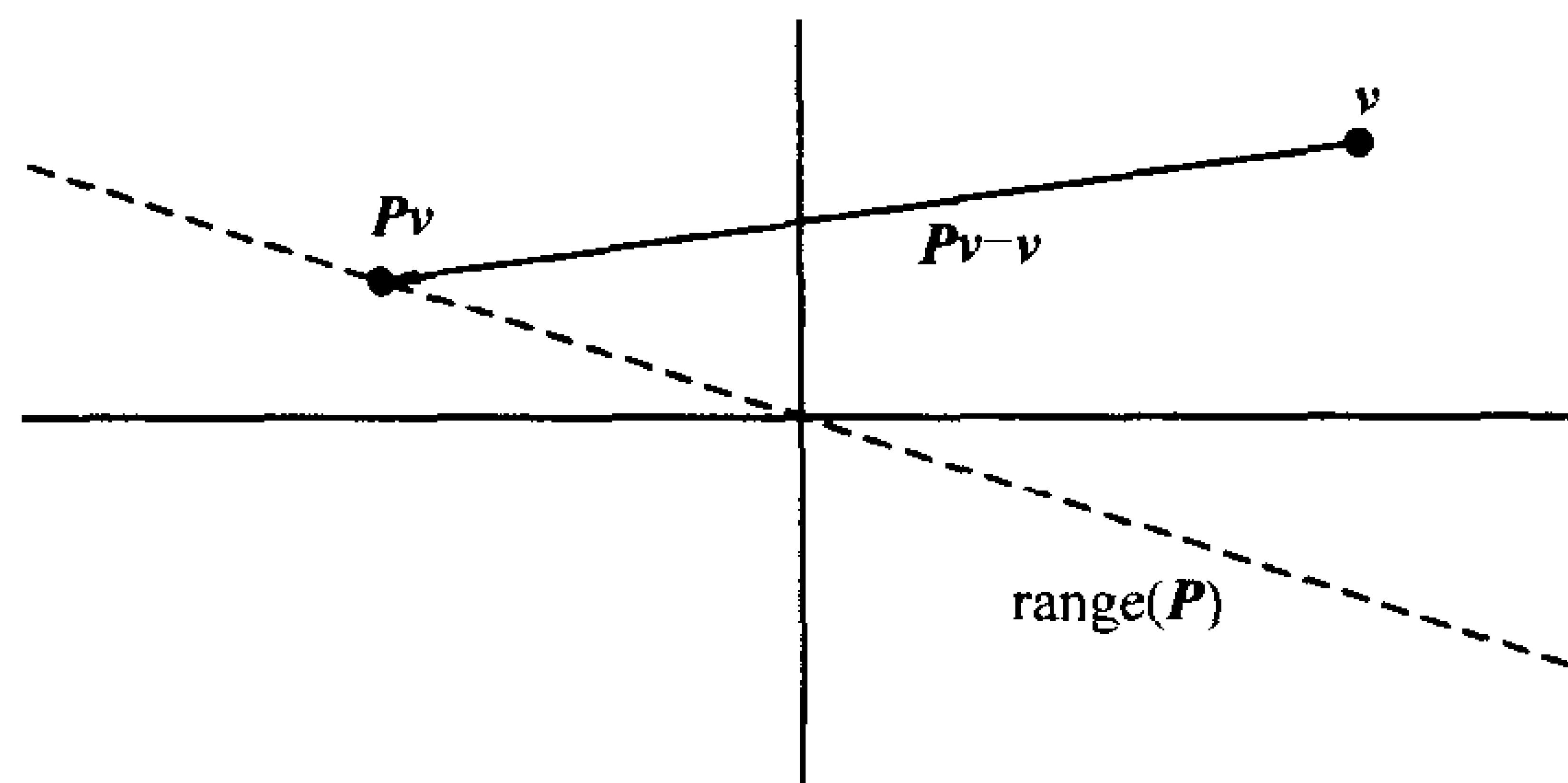


图6-1 一个斜投影

6.2 互补投影算子

若P是一个投影算子，则$I-P$也是投影算子，因为它也是幂等的：

$$(I-P)^2=I-2P+P^2=I-P.$$

矩阵$I-P$称为P的*互补投影算子*（complementary projector）.

$I-P$投影到什么空间上？准确地说是P的零空间！我们知道$\operatorname{range}(I-P)\supseteq\operatorname{null}(P)$，因为若$Pv=0$，有$(I-P)v=v$. 反之，我们知道$\operatorname{range}(I-P)\subseteq\operatorname{null}(P)$，因为对任意的$v$，有$(I-P)v=v-Pv\in\operatorname{null}(P)$. 所以，对任意的投影算子$P$，

$$\operatorname{range}(I-P)=\operatorname{null}(P). \tag{6.2}$$

将P写成$P=I-(I-P)$，可以写出与（6.2）互补的事实：

$$\operatorname{null}(I-P)=\operatorname{range}(P). \tag{6.3}$$

还可以看到，$\operatorname{null}(I-P)\cap\operatorname{null}(P)=\{0\}$：在这两个集合的向量$v$满足$v=v-Pv=(I-P)v=0$. 叙述这个结论的另一方式是

$$\operatorname{range}(P)\cap\operatorname{null}(P)=\{0\}. \tag{6.4}$$

这些计算表明，一个投影算子将$\mathbb{C}^m$分离为两个空间. 反之，令S_1和S_2为$\mathbb{C}^m$的两个子空间，满足$S_1\cap S_2=\{0\}$和$S_1+S_2=\mathbb{C}^m$，其中S_1+S_2记为S_1和S_2张成的空间，即向量s_1+s_2的集合，其中$s_1\in S_1$，$s_2\in S_2$. [这样的一对子空间称为*互补的子空间*（complementary subspace）.] 这样，存在一个投影算子P，使得$\operatorname{range}(P)=S_1$及$\operatorname{null}(P)=S_2$. 称$P$是*沿$S_2$到$S_1$上*的投影算子，这个投影算子和它的补可以看成下列问题的惟一解：

42

$$\text{给定 } v\text{，求向量 } v_1\in S_1 \text{ 和 } v_2\in S_2\text{，使得 } v_1+v_2=v.$$

投影Pv给出v_1，而补投影$(I-P)v$给出v_2，这些向量是惟一的，因为所有的解一定有形式

$$(Pv+v_3)+((I-P)v-v_3)=v,$$

其中，显然v_3一定既在S_1也在S_2中，即$v_3=0$.

在一种情形下，投影算子和它们补算子的产生使人感到特别熟悉. 假设$m\times m$矩阵A有一个完备的特征向量集$\{v_j\}$，如（5.1）所示，意即$\{v_j\}$是$\mathbb{C}^m$的一组基. 我们频繁地将问题与向量在这组基中的展开式联系起来，例如，令$x\in\mathbb{C}^m$，x在一个特殊的特征向量方向的分量是什么？答案是Px，其中P是某个秩1投影算子. 然而，我们这里并不给出详细的讨论，而是转到投影算子的特殊类别上，对我们来说，这才是本书中主要的兴趣所在.

6.3 正交投影算子

正交投影算子（orthogonal projector）（图6-2）是一个沿着空间 S_2 投影到子空间 S_1 上的投影算子，其中 S_1 和 S_2 是正交的.（警告：正交投影算子并非正交矩阵！）

43

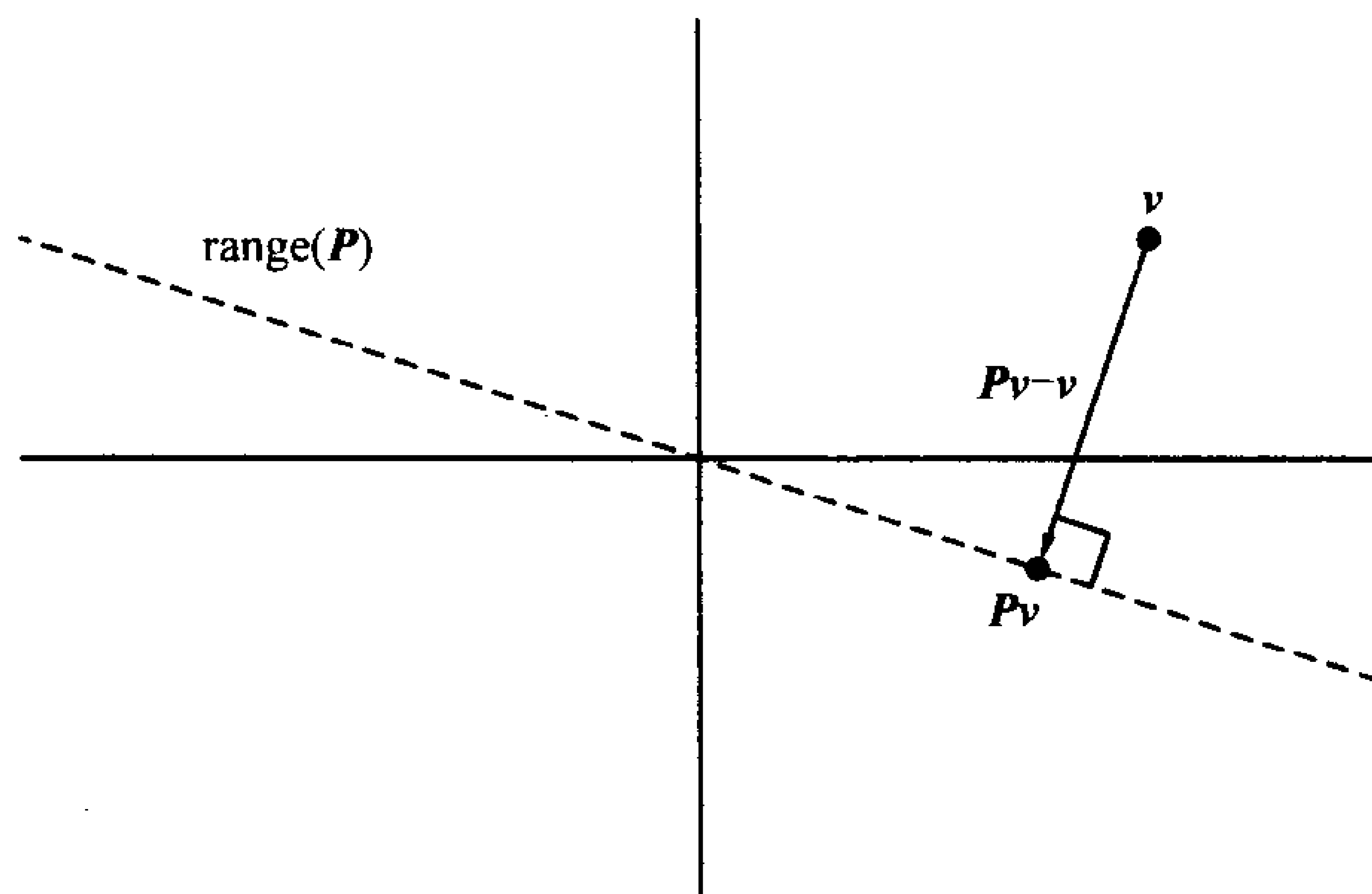


图6-2 一个正交投影

也有一个代数的定义：正交投影算子是这样的投影算子，它是埃米尔特矩阵，除了满足（6.1）之外也满足 $P^* = P$. 当然，我们必须要证明这个定义与开头的定义是等价的.

定理6.1 一个投影算子是正交的当且仅当 $P = P^*$.

证明 若 $P = P^*$，则向量 $Px \in S_1$ 和向量 $(I - P)y \in S_2$ 之间的内积为零：

$$x^* P^* (I - P)y = x^* (P - P^2)y = 0.$$

因此投影算子是正交的，这给出了定理中“当”的证明.

对于“仅当”的证明，可以利用SVD. 假定 P 沿 S_2 投影到 S_1 上，其中 $S_1 \perp S_2$ 且 S_1 有维数 n，则 P 的一个SVD可如下构造. 令 $\{q_1, q_2, \cdots, q_m\}$ 为 $\mathbb{C}^m$ 的一组正交基，其中 $\{q_1, \cdots, q_n\}$ 为 S_1 的一组基且 $\{q_{n+1}, \cdots, q_m\}$ 为 S_2 的一组基. 对 $j \leqslant n$，有 $Pq_j = q_j$，而对 $j > n$，有 $Pq_j = 0$. 现令 Q 为第 j 列是 q_j 的酉矩阵，则有

$$PQ = \left[\begin{array}{c|c|c|c|c} q_1 & \cdots & q_n & 0 & \cdots \end{array} \right],$$

所以，有

$$Q^*PQ = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & 0 & \\ & & & & \ddots \end{bmatrix} = \Sigma,$$

它是前 n 个对角元素为 1 而其他对角元素为零的对角矩阵. 因此我们已经构造了 P 的一个奇异值分解：

$$P = Q\Sigma Q^*. \tag{6.5}$$

（注意，这也是一个特征值分解（5.1).）由此可见，因为 $P^* = (Q\Sigma Q^*)^* = Q\Sigma^* Q^* = Q\Sigma Q^* = P$，所以 P 是埃米尔特矩阵. □

44

6.4　利用正交基的投影

因为一个正交投影算子有某些奇异值等于零（除了平凡情形 $P = I$)，自然会在（6.5）中去掉 Q 的不起作用的列，且用约化的 SVD 而非全 SVD，得到相当简单的表达式

$$P = \hat{Q}\hat{Q}^*, \tag{6.6}$$

其中，$\hat{Q}$的列是正交的.

在（6.6）中，矩阵$\hat{Q}$不必来自 SVD. 令$\{q_1, \cdots, q_n\}$为$\mathbb{C}^m$中任意 n 个正交向量的集合，令$\hat{Q}$为其对应的 $m \times n$ 矩阵. 由（2.7）可知

$$v = r + \sum_{i=1}^{n} (q_i q_i^*) v$$

表示一个向量 $v \in \mathbb{C}^m$ 分解到$\hat{Q}$的列空间的分量加上一个在正交空间的分量，因此映射

$$v \mapsto \sum_{i=1}^{n} (q_i q_i^*) v \tag{6.7}$$

是一个在 range($\hat{Q}$) 上的正交投影算子，用矩阵形式可写成 $y = \hat{Q}\hat{Q}^* v$，

$$\underset{y}{\boxed{\;}} = \underset{\hat{Q}}{\boxed{\quad}}\;\underset{\hat{Q}^*}{\boxed{\quad}}\;\underset{v}{\boxed{\;}}.$$

因而不管$\hat{Q}$是怎样得到的，只要它的列是正交的，任意的乘积$\hat{Q}\hat{Q}^*$总是一个在$\hat{Q}$的列空间上的投影算子. 也许$\hat{Q}$可由单位矩阵的全因子分解 $v = QQ^*v$ 中删去某些行和列得到，也许并非如此.

$$v = Q \quad Q^* \quad v,$$

正交算子的补也是正交算子（证明：$I-\hat{Q}\hat{Q}^*$ 是埃米尔特矩阵），补算子投影到正交于 range($\hat{Q}$) 的空间上.

45

正交算子的一个重要特殊情形是秩 1 正交投影算子，它把分量分隔在一个单独的方向 q 中，这可写成

$$P_q = qq^*. \tag{6.8}$$

这些是由在（6.7）中所示的高秩投影算子所得结果的一些片段. 它们的补算子是消去 q 方向分量的秩 $m-1$ 正交投影算子：

$$P_{\perp q} = I - qq^*. \tag{6.9}$$

方程（6.8）和（6.9）中假设 q 是单位向量，对任意的非零向量 a，类似的公式是

$$P_a = \frac{aa^*}{a^*a}, \tag{6.10}$$

$$P_{\perp a} = I - \frac{aa^*}{a^*a}. \tag{6.11}$$

6.5 利用任意基的投影

在$\mathbb{C}^m$ 的子空间上的正交投影算子也可以用任意的基开始构造，而不必是正交的基. 假设子空间由线性无关向量$\{a_1,\cdots,a_n\}$所张成，且令 A 为第 j 列是 a_j 的 $m\times n$ 矩阵.

以前所述的由 v 到其正交投影 $y\in\text{range}(A)$，其差 $y-v$ 一定正交于 range(A)，这等价于说，y 一定满足对每个 j，有 $a_j^*(y-v)=0$. 因为 $y\in\text{range}(A)$，可令 $y=Ax$，将上述条件写成 $a_j^*(Ax-v)=0$ 对每个 j 成立，或等价地，$A^*(Ax-v)=0$ 或 $A^*Ax=A^*v$. 易见，因 A 满秩，所以 A^*A 非奇异（习题6.3）. 因此

$$x = (A^*A)^{-1}A^*v. \tag{6.12}$$

最后，v 的投影，$y = Ax$ 为 $y = A(A^*A)^{-1}A^*v$. 因此投影到 range(A)上的正交投影算子可表示为

$$P = A(A^*A)^{-1}A^*. \tag{6.13}$$

注意，这是（6.10）的多维推广. 在正交情形 $A = \hat{Q}$中，括号中的项化为单位矩阵，重新复原为（6.6）.

46

习　　题

6.1　若 P 为正交投影算子，则 $I - 2P$ 是酉的. 用代数方法证明这个结论并给出几何解释.

6.2　令 E 为 $m \times m$ 矩阵，它提取一个 m 维向量的“偶部分”：$Ex = (x + Fx)/2$，其中 F 是一个翻转$(x_1, \cdots, x_m)^*$为$(x_m, \cdots, x_1)^*$的 $m \times m$ 矩阵. E 是一个正交投影算子，还是一个斜投影算子，或者根本不是投影算子？它的元素是什么？

6.3　对 $m \geqslant n$，给定 $A \in \mathbb{C}^{m \times n}$，证明 A^*A 非奇异当且仅当 A 满秩.

6.4　考虑矩阵

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

用手算回答下列问题.

(a) 什么是投影到 range(A)上的正交投影算子 P，什么是 P 作用在向量（1，2，3）*上的像？

(b) 对 B 做同样的问题.

6.5　令 $P \in \mathbb{C}^{m \times m}$ 为一非零投影算子，证明 $\|P\|_2 \geqslant 1$，等号当且仅当 P 为正交投影算子时成立.

47

第 7 讲　QR 因子分解

在数值线性代数中，有一种算法的思想比起其他所有算法的思想更为重要，这就是 QR 因子分解.

7.1　约化 QR 因子分解

对于很多应用，我们发现自己的兴趣在于矩阵 A 的列空间系列. 请注意这里所说的空间是一系列的，这些是由 A 的列 $a_1, a_2, \cdots$ 逐个张成的空间系：

$$\langle a_1 \rangle \subseteq \langle a_1, a_2 \rangle \subseteq \langle a_1, a_2, a_3 \rangle \subseteq \cdots$$

这里与第 5 讲及全书一样，记号〈…〉指由括号内所含向量张成的子空间. 因此 $\langle a_1 \rangle$ 是由 a_1 张成的一维空间，$\langle a_1, a_2 \rangle$ 是由 a_1 和 a_2 张成的二维空间，如此等等. QR 因子分解的思想就在于张成这些逐个空间的正交向量序列 $q_1, q_2, \cdots$ 的构造.

精确地说，目前暂假设 $A \in \mathbb{C}^{m\times n}$（$m \geqslant n$）有满秩 n. 我们希望序列 $q_1, q_2, \cdots$ 具有性质

$$\langle q_1, q_2, \cdots, q_j \rangle = \langle a_1, a_2, \cdots, a_j \rangle, \quad j = 1, \cdots, n. \tag{7.1}$$

从第 1 讲所述，不难看到这相当于条件

$$\left[\begin{array}{c|c|c|c} & & & \\ a_1 & a_2 & \cdots & a_n \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ q_1 & q_2 & \cdots & q_n \\ & & & \end{array}\right] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & \vdots \\ & & \ddots & r_{nn} \end{bmatrix}, \tag{7.2}$$

其中对角元素 r_{kk} 非零. 因若（7.2）成立，则 $a_1, \cdots, a_k$ 可表示为 $q_1, \cdots, q_k$ 的线性组合，且三角矩阵的左上角 $k \times k$ 块的可逆性隐含其中，反之，$q_1, \cdots, q_k$ 可表示为 $a_1, \cdots, a_k$ 的线性组合. 将其写出，这些方程有如下形式

$$\begin{aligned} a_1 &= r_{11} q_1, \\ a_2 &= r_{12} q_1 + r_{22} q_2, \\ a_3 &= r_{13} q_1 + r_{23} q_2 + r_{33} q_3, \\ &\vdots \\ a_n &= r_{1n} q_1 + r_{2n} q_2 + \cdots + r_{nn} q_n. \end{aligned} \tag{7.3}$$

用矩阵表示有

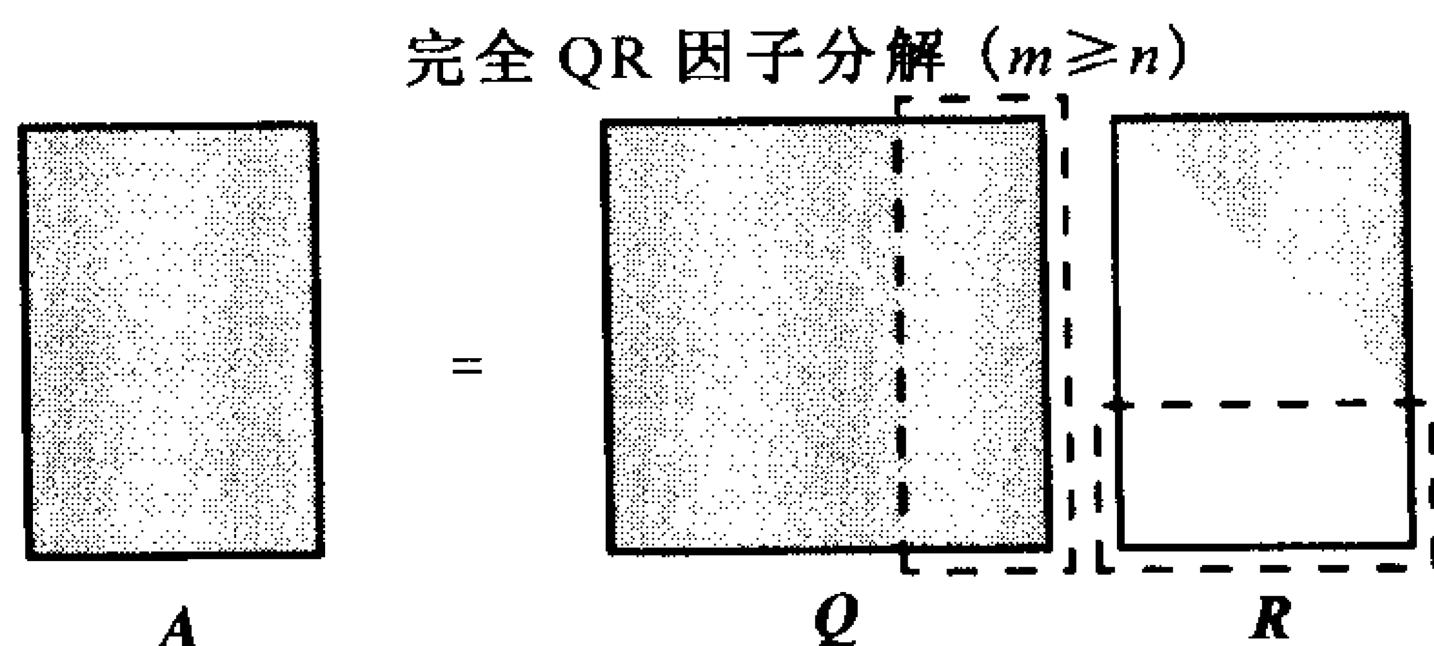
$$A = \hat{Q}\hat{R}, \tag{7.4}$$

其中，$\hat{Q}$为具有正交列的 $m\times n$ 矩阵，$\hat{R}$为 $n\times n$ 的上三角矩阵，这样的因子分解称为 A 的约化 QR 因子分解（reduced QR factorization of A）.

7.2 完全 QR 因子分解

对$\hat{Q}$附上外加的 $m-n$ 列正交列，使它成为一个 $m\times m$ 的酉矩阵 Q，就得到 $A\in\mathbb{C}^{m\times n}$ $(m\geqslant n)$的完全 QR 因子分解（full QR factorization），这类似于第 4 讲中所叙述的由约化到完全 SVD 的一段. 在这个过程中，零元素的行附加到$\hat{R}$上使它成为一个 $m\times n$矩阵 R，它仍然是上三角的. 完全与约化 QR 因子分解之间的关系如下所示.

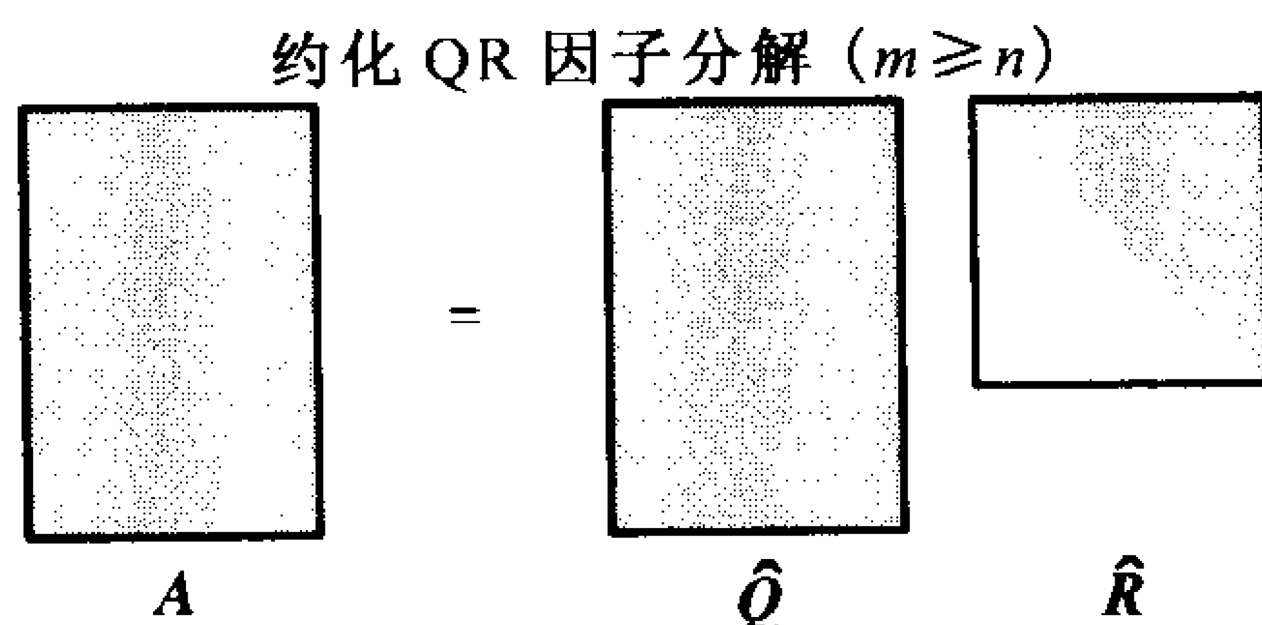
完全 QR 因子分解 $(m\geqslant n)$



49

在完全 QR 因子分解中，Q 是 $m\times m$ 的，R 是 $m\times n$ 的，Q 的最后 $m-n$ 列乘以 R 中的零（虚线所围）. 在约化 QR 因子分解中，不起作用的列和行都删去了. 现在$\hat{Q}$是 $m\times n$ 的，$\hat{R}$是 $n\times n$ 的，而在$\hat{R}$中没有的行必须为零.

约化 QR 因子分解 $(m\geqslant n)$



注意在完全 QR 因子分解中，对 $j>n$，列 q_j 正交于 range(A). 设 A 有满秩 n，它们构成了 range$(A)^{\perp}$（正交于 range(A)的空间）的正交基，或等价地说，是 null(A^*)的正交基.

7.3 格拉姆-施密特正交化

方程组（7.3）提示了计算约化 QR 因子分解的一个方法. 给定 $a_1,a_2,\cdots$，可用逐次正交化的过程构造向量 $q_1,q_2,\cdots$及元素 r_{ij}. 这是一种历史久远的思想，称为格拉姆-施密特正交化（Gram-Schmidt orthogonalization）.

计算过程如下，在第 j 步，希望找到一个单位向量 $q_j\in\langle a_1,\cdots,a_j\rangle$正交于 $q_1,\cdots,q_{j-1}$. 这样做时，我们已经考虑了所需的在（2.6）中的正交化技术，由该方程看到

$$v_j = a_j - (q_1^* a_j)q_1 - (q_2^* a_j)q_2 - \cdots - (q_{j-1}^* a_j)q_{j-1} \tag{7.5}$$

是所需类型的向量，只是还没有规范化. 如果除以$\|v_j\|_2$，那么结果就是合适的向量q_j.

根据这样的想法，将（7.3）重新写成形式

$$\begin{aligned} q_1 &= \frac{a_1}{r_{11}}, \\ q_2 &= \frac{a_2 - r_{12}q_1}{r_{22}}, \\ q_3 &= \frac{a_3 - r_{13}q_1 - r_{23}q_2}{r_{33}}, \\ &\vdots \\ q_n &= \frac{a_n - \sum_{i=1}^{n-1} r_{in}q_i}{r_{nn}}. \end{aligned} \tag{7.6}$$

50

由（7.5），显然对（7.6）中分子的系数r_{ij}适当的定义为

$$r_{ij} = q_i^* a_j \quad (i \neq j). \tag{7.7}$$

为了规范化，分母中的系数r_{jj}选为：

$$|r_{jj}| = \left\| a_j - \sum_{i=1}^{j-1} r_{ij} q_i \right\|_2. \tag{7.8}$$

注意，r_{jj}的符号是不确定的. 随意地可选$r_{jj}>0$，在这种情形中，我们将完成$\hat{R}$沿对角线有正元素的因子分解$A=\hat{Q}\hat{R}$.

(7.6)～(7.8) 所概括的算法就是格拉姆-施密特迭代. 从数学方面说，它给出了一个简单程序去理解和证明 QR 因子分解的各种性质. 从数值计算方面说，因为计算机的舍入误差，它是不稳定的. 为了强调不稳定性，数值分析称它为**经典格拉姆-施密特迭代**（classical Gram-Schmidt iteration），与之相对立的是**修正的格拉姆-施密特迭代**（modified Gram-Schmidt iteration），这将在下一讲中讨论.

算法 7.1　经典格拉姆-施密特（不稳定）

```
for j = 1 to n
    v_j = a_j
    for i = 1 to j-1
        r_ij = q_i^* a_j
        v_j = v_j - r_ij q_i
    r_jj = ||v_j||_2
    q_j = v_j / r_jj
```

7.4 存在性和惟一性

所有矩阵都有QR因子分解，在适当的条件限制下，它们是惟一的. 我们首先建立存在性的结论.

定理 7.1 *每个$A\in\mathbb{C}^{m\times n}(m\geq n)$有完全QR因子分解，因而也有约化QR因子分解.*

证明 首先假设A满秩且只讨论约化QR因子分解，在此情形中，存在性的证明由格拉姆-施密特算法本身给出. 由算法的构造，其过程产生$\hat{Q}$的正交列及$\hat{R}$的元素，使得 [51] (7.4) 成立. 失败只能在某步v_j为零时出现，此时不能规范化产生q_j. 然而，这会隐含$a_j\in\langle q_1,\cdots,q_{j-1}\rangle=\langle a_1,\cdots,a_{j-1}\rangle$，与$A$满秩矛盾.

现在假设A非满秩，则在一步或多步j中，将会发现 (7.5) 给出$v_j=0$，如刚才指出的那样. 此时我们简单地任取q_j为任何正交于$\langle q_1,\cdots,q_{j-1}\rangle$的规范化向量，并继续格拉姆-施密特过程.

最后，除了约化情形外，$m\times n$矩阵 ($m>n$) 的完全QR因子分解可以用同样的方式引入任意正交向量来构造. 进行格拉姆-施密特过程到第n步，然后继续到附加的$m-n$步，在每步中引入向量q_j.

最后两段讨论的结果已经在第4讲中关于SVD的讨论中提到过. □

现在转向惟一性. 设$A=\hat{Q}\hat{R}$是一个约化QR因子分解，对某个满足$|z|=1$的数量z，如果$\hat{Q}$的第i列乘以z以及$\hat{R}$的第i行乘以z^{-1}，得到A的另一个QR因子分解. 下面的定理断言，若A满秩，这是获得不同约化QR因子分解的惟一途径.

定理 7.2 *每个满秩矩阵$A\in\mathbb{C}^{m\times n}(m\geq n)$有惟一的约化QR因子分解$A=\hat{Q}\hat{R}$，其中$r_{jj}>0$.*

证明 证明再次由格拉姆-施密特迭代给出. 由式 (7.4)、$\hat{Q}$列的正交性以及$\hat{R}$的上三角性质得，

A的任意约化QR因子分解一定满足 (7.6) ~ (7.8). 由满秩的假设，(7.6) 中的分母 (7.8) 非零，因而在每个逐次的步j，这些公式完全确定了r_{ij}和q_j，只除了一处：在 (7.8) 中，r_{jj}的符号未确定. 一旦像算法7.1那样，由条件$r_{jj}>0$固定，因子分解便完全确定. □

7.5 向量变成连续函数的情形

对于函数（而不是向量）的正交展开，有类似的QR因子分解.

设将$\mathbb{C}^m$用L^2 [-1, 1] 代替，它是 [-1, 1] 上复值函数的向量空间. 我们

不正式地引入这个空间的性质，只指出 f 和 g 的内积，取为

$$(f,g)=\int_{-1}^{1}\overline{f(x)}g(x)\mathrm{d}x. \tag{7.9}$$

52

就够了，例如考虑下面其“列”为单项式 x^j 的“矩阵”：

$$A=\left[\begin{array}{c|c|c|c|c} \\ 1 & x & x^2 & \cdots & x^{n-1} \\ \\ \end{array}\right]. \tag{7.10}$$

它的每一列是 $L^2[-1,1]$ 中的一个函数，因而，通常 A 在水平方向上是离散的，而在垂直方向上是连续的. 它是例 1.1 中范德蒙德矩阵（1.4）的连续模拟.

A 的“连续 QR 因子分解”形为

$$A=QR=\left[\begin{array}{c|c|c|c} \\ q_0(x) & q_1(x) & \cdots & q_{n-1}(x) \\ \\ \end{array}\right]\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & \vdots \\ & & \ddots & r_{nn} \end{bmatrix},$$

其中，Q 的列是 x 的函数，关于内积（7.9）正交：

$$\int_{-1}^{1}\overline{q_i(x)}q_j(x)\mathrm{d}x=\delta_{ij}=\begin{cases}1 & 若\ i=j,\\ 0 & 若\ i\neq j.\end{cases}$$

由格拉姆-施密特构造可见，q_j 是一个 j 次多项式这些多项式是所谓的*勒让德多项式*（Legendre polynomial）P_j 的数乘，其中 P_j 按习惯规范化为 $P_j(1)=1$. 前几个 P_j 是

$$P_0(x)=1, P_1(x)=x, P_2(x)=\frac{3}{2}x^2-\frac{1}{2}, P_3(x)=\frac{5}{2}x^3-\frac{3}{2}x; \tag{7.11}$$

见图 7-1. 像单项式 $1,x,x^2,\cdots$ 那样，这个多项式的序列逐步张成更高次的多项式空间. 然而，$P_0(x),P_1(x),P_2(x),\cdots$ 具有相互正交的优点，这使得它们更适用于某些计算. 事实上，利用这种多项式的计算构成了*谱方法*（spectral method）的基础，这是偏微分方程数值解法中最有效的技术之一.

什么是与 $\hat{Q}$ 联系的“投影矩阵”$\hat{Q}\hat{Q}^*$（6.6）? 它是一个“$[-1,1]\times[-1,1]$ 矩阵”，即一个积分算子

$$f(\cdot)\mapsto\sum_{j=0}^{n-1}q_j(\cdot)\int_{-1}^{1}\overline{q_j(x)}f(x)\mathrm{d}x \tag{7.12}$$

它把 $L^2[-1,1]$ 中的函数映射到 $L^2[-1,1]$ 中的函数.

53

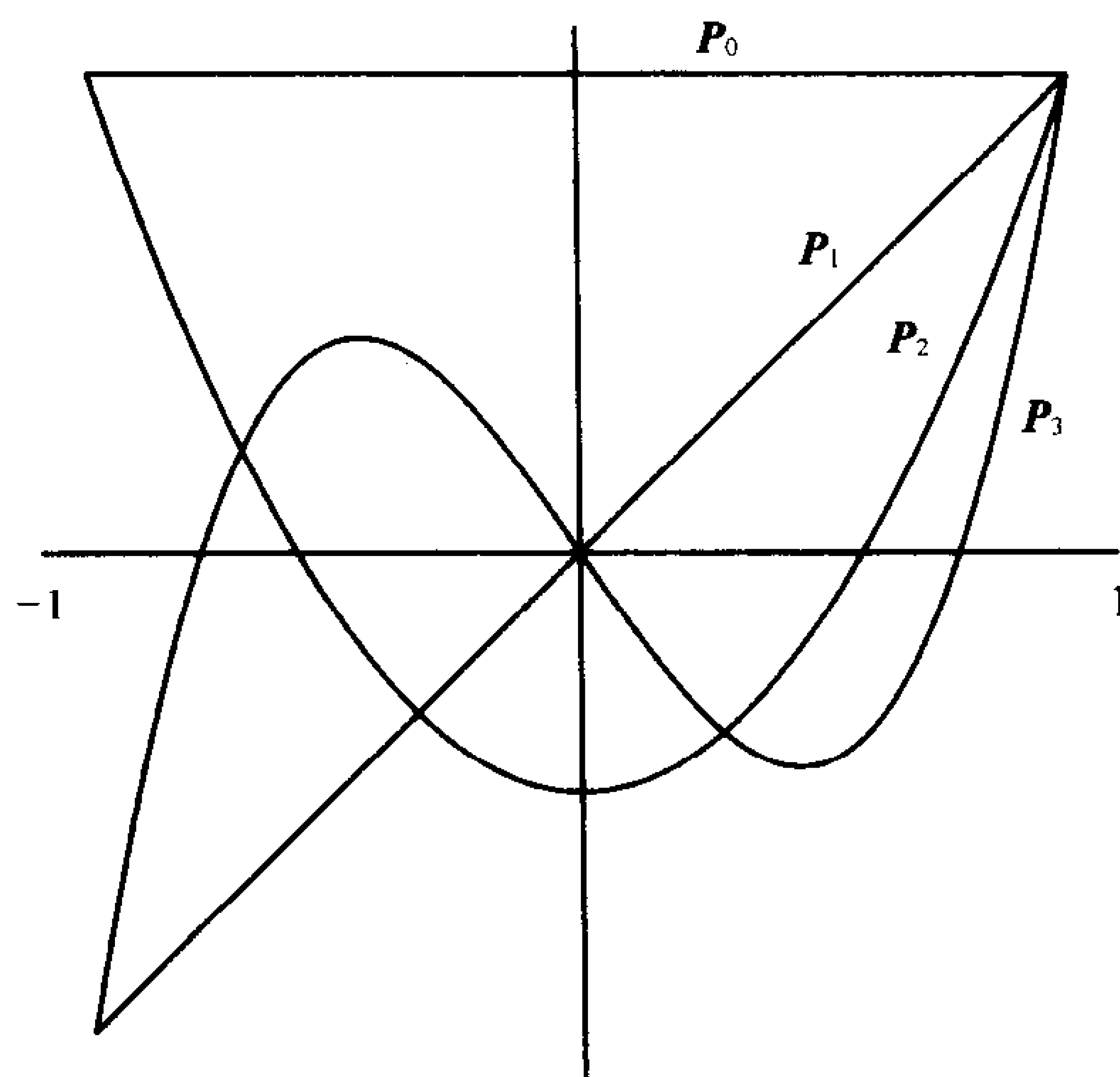


图 7-1　前 4 个勒让德多项式（7.11），除去数量因子，这些函数可以解释为，在一个“$[-1,1]\times 4$ 矩阵”$[1,x,x^2,x^3]$的约化 QR 因子分解中$\hat{Q}$的列

7.6　用 QR 因子分解解方程组 $Ax=b$

在本讲的末尾，转回到离散、有限的矩阵. 假设要对 x 解 $Ax=b$，其中$A\in\mathbb{C}^{m\times m}$非奇异，若 $A=QR$ 是一个 QR 因子分解，则方程组可写成 $QRx=b$ 或

$$Rx = Q^*b. \tag{7.13}$$

若 Q 已知，方程的右边易于计算，而线性方程组的左边是三角形的，它也容易求解. 这提示了下述解 $Ax=b$ 的计算方法

1. 计算 QR 因子分解 $A=QR$.
2. 计算 $y=Q^*b$.
3. 对 x 解 $Rx=y$.

在后面的几讲中，将给出这些步骤中每步的算法.

以上的步骤是解线性方程组一种很好的方法；在第 16 讲将证明这一点. 然而它不是这种问题的标准方法. 在实际中一般用高斯消去法，因为它只需一半的数值运算量.

54

习　题

7.1　再次考虑习题 6.4 的矩阵 A 和 B.

(a) 用你喜欢的任何方法，在纸上确定 A 的一个约化 QR 因子分解 $A=\hat{Q}\hat{R}$和完全 QR 因子分解 $A=QR$.

(b) 再次用你喜欢的任何方法，确定约化和完全的 QR 因子分解 $B=\hat{Q}\hat{R}$和 $B=QR$.

7.2 令矩阵 A 满足：其 1,3,5,7,…列正交于 2,4,6,8…列，在约化 QR 因子分解 $A=\hat{Q}\hat{R}$中，$\hat{R}$有什么特殊的结构？可以假设 A 满秩.

7.3 令 A 为 $m\times m$ 矩阵，a_j 为它的第 j 列，给出阿达马不等式

$$|\det A| \leqslant \prod_{j=1}^{m} \|a_j\|_2$$

的一个代数证明，同时给出这个结论的一个几何解释，用行列式等于平行六面体体积的事实.

7.4 令 $x^{(1)}$，$y^{(1)}$，$x^{(2)}$和 $y^{(2)}$为$\mathbb{R}^3$ 中的非零向量，其中 $x^{(1)}$和 $y^{(1)}$线性无关，$x^{(2)}$和 $y^{(2)}$也线性无关. 考虑$\mathbb{R}^3$ 中的两个平面

$$P^{(1)}=\langle x^{(1)},y^{(1)}\rangle,\quad P^{(2)}=\langle x^{(2)},y^{(2)}\rangle.$$

假如希望在交集 $P=P^{(1)}\cap P^{(2)}$中找到一个非零向量 $v\in\mathbb{R}^3$，设计解此问题的一个方法，其中利用把问题化为 3 个 3×2 矩阵的 QR 因子分解的计算.

7.5 令 A 为 $m\times n$ 矩阵（$m\geqslant n$），且令 $A=\hat{Q}\hat{R}$为一个约化 QR 因子分解.

(a) 证明 A 有秩 n 当且仅当$\hat{R}$的所有对角元素非零.

(b) 假设对某个满足 $0\leqslant k<n$ 的 k，$\hat{R}$有 k 个非零对角元素. 这隐含了关于 A 的秩的什么性质？准确地说是 k？至少是 k？至多是 k？给出精确的答案，并证明它.

55

第 8 讲　格拉姆-施密特正交化

格拉姆-施密特迭代是计算 QR 因子分解的两个主要数值算法之一的基础，它是一个“三角正交化”的过程，通过系列的矩阵运算使矩阵的列正交，这可解释为右乘上三角矩阵的乘法.

8.1　格拉姆-施密特投影

上一讲给出了经典形式的格拉姆-施密特迭代. 在本讲开头，我们用另外一个利用正交投影算子的方法再次描述同样的算法.

令 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$ $(m\geqslant n)$ 是列为$\{\boldsymbol{a}_j\}$的满秩矩阵，先前，我们用公式 (7.6) ~ (7.8) 表示格拉姆-施密特迭代. 现在考虑系列公式

$$\boldsymbol{q}_1=\frac{\boldsymbol{P}_1\boldsymbol{a}_1}{\|\boldsymbol{P}_1\boldsymbol{a}_1\|},\ \boldsymbol{q}_2=\frac{\boldsymbol{P}_2\boldsymbol{a}_2}{\|\boldsymbol{P}_2\boldsymbol{a}_2\|},\ \cdots,\ \boldsymbol{q}_n=\frac{\boldsymbol{P}_n\boldsymbol{a}_n}{\|\boldsymbol{P}_n\boldsymbol{a}_n\|}. \tag{8.1}$$

在这些公式中，每个 $\boldsymbol{P}_j$ 记为一个正交投影算子，具体地说，$\boldsymbol{P}_j$ 是秩 $m-(j-1)$ 的 $m\times n$矩阵，它正交地投影$\mathbb{C}^m$ 到$\langle\boldsymbol{q}_1,\cdots,\boldsymbol{q}_{j-1}\rangle$的正交空间. (在 $j=1$ 情形，此规定指的是恒等：$\boldsymbol{P}_1=\boldsymbol{I}$.) 现在，显然由 (8.1) 定义的 $\boldsymbol{q}_j$ 正交于 $\boldsymbol{q}_1,\cdots,\boldsymbol{q}_{j-1}$，它在空间$\langle\boldsymbol{a}_1,\cdots,\boldsymbol{a}_j\rangle$内，且范数为 1，因此可见 (8.1) 等价于 (7.6) ~ (7.8) 及算法 7.1.

可以写出投影算子 $\boldsymbol{P}_j$ 的显示表达式. 令$\hat{\boldsymbol{Q}}_{j-1}$记为包含$\hat{\boldsymbol{Q}}$前 $j-1$ 列的 $m\times(j-1)$矩阵，

$$\hat{\boldsymbol{Q}}_{j-1}=\left[\begin{array}{c|c|c|c} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_{j-1} \end{array}\right]. \tag{8.2}$$

则 $\boldsymbol{P}_j$ 由

$$\boldsymbol{P}_j=\boldsymbol{I}-\hat{\boldsymbol{Q}}_{j-1}\hat{\boldsymbol{Q}}_{j-1}^{*}. \tag{8.3}$$

给出. 现在，读者可能对我们的记号和正交性的思想已足够熟悉，以致一下子就看出，(8.3) 表示 (7.5) 中应用到 $\boldsymbol{a}_j$ 的算子.

8.2　修正格拉姆-施密特算法

正如我们曾指出的那样，实际上在算法 7.1 和 (8.1) 的格拉姆-施密特公式是不能应用的，因为这系列的计算导致数值不稳定，幸运的是，一个简单的修正使问

题得到改进，我们仍然尚未讨论数值的稳定性；这将在下一讲中涉及并在第 14 讲中开始系统地讨论. 目前，知道一个稳定的对计算机舍入误差的影响不太灵敏的算法，这就足够了.

对每个 j 的值，算法 7.1 计算单个秩 $m-(j-1)$ 的正交投影

$$\boldsymbol{v}_j = \boldsymbol{P}_j \boldsymbol{a}_j. \tag{8.4}$$

相反的是，修正格拉姆-施密特算法由 $j-1$ 个秩 $m-1$ 投影的序列计算同样的结果. 回顾（6.9），$\boldsymbol{P}_{\perp q}$ 记为投影在正交于非零向量 $\boldsymbol{q}\in\mathbb{C}^m$ 的空间上秩 $m-1$ 的正交投影算子. 由 $\boldsymbol{P}_j$ 的定义，不难看到

$$\boldsymbol{P}_j = \boldsymbol{P}_{\perp q_{j-1}} \cdots \boldsymbol{P}_{\perp q_2} \boldsymbol{P}_{\perp q_1}, \tag{8.5}$$

并再次用 $\boldsymbol{P}_1 = \boldsymbol{I}$. 因此，（8.4）的一个等价陈述为

$$\boldsymbol{v}_j = \boldsymbol{P}_{\perp q_{j-1}} \cdots \boldsymbol{P}_{\perp q_2} \boldsymbol{P}_{\perp q_1} \boldsymbol{a}_j. \tag{8.6}$$

修正格拉姆-施密特算法是基于用（8.6）替代（8.4）的算法.　57

从数学上看，（8.6）和（8.4）是等价的. 然而，隐含在这些公式中的系列算术运算是不同的，修正算法以依次给出下列公式的值来计算 $\boldsymbol{v}_j$：

$$\begin{aligned}
\boldsymbol{v}_j^{(1)} &= \boldsymbol{a}_j, \\
\boldsymbol{v}_j^{(2)} &= \boldsymbol{P}_{\perp q_1}\boldsymbol{v}_j^{(1)} = \boldsymbol{v}_j^{(1)} - \boldsymbol{q}_1\boldsymbol{q}_1^*\boldsymbol{v}_j^{(1)}, \\
\boldsymbol{v}_j^{(3)} &= \boldsymbol{P}_{\perp q_2}\boldsymbol{v}_j^{(2)} = \boldsymbol{v}_j^{(2)} - \boldsymbol{q}_2\boldsymbol{q}_2^*\boldsymbol{v}_j^{(2)}, \\
&\ \vdots \qquad\qquad \vdots \\
\boldsymbol{v}_j = \boldsymbol{v}_j^{(j)} &= \boldsymbol{P}_{\perp q_{j-1}}\boldsymbol{v}_j^{(j-1)} = \boldsymbol{v}_j^{(j-1)} - \boldsymbol{q}_{j-1}\boldsymbol{q}_{j-1}^*\boldsymbol{v}_j^{(j-1)}.
\end{aligned} \tag{8.7}$$

在有限精度的计算机运算中，可以看到，（8.7）比（8.4）引入的误差要小.

当算法实现时，在 $\boldsymbol{q}_i$ 已知之后，对每个 $j>i$，投影算子 $\boldsymbol{P}_{\perp q_i}$ 可以立刻方便地应用到 $\boldsymbol{v}_j^{(i)}$ 上. 下面的描述就是这样的.

算法 8.1　修正格拉姆-施密特

for $i=1$ **to** n
　　$v_i = a_i$
for $i=1$ **to** n
　　$r_{ii} = \|v_i\|$
　　$q_i = v_i / r_{ii}$
　　for $j=i+1$ **to** n
　　　　$r_{ij} = q_i^* v_j$
　　　　$v_j = v_j - r_{ij} q_i$

实际上，为了节约存储，令 v_i 覆盖 a_i，同样 q_i 覆盖 v_i.

读者可以比较算法7.1和8.1，直到能确信两者的等价性.

8.3 运算计数

格拉姆-施密特算法是本书给出的第一个算法，对于任何算法，估计它的代价是重要的. 为此，贯穿全书我们沿用经典的方法，并且计算算法所需的浮点运算——"flop"的总数. 每个加法、减法、乘法、除法或平方根计作一次浮点运算，我们不区别实运算和复运算，虽然实际上大多数计算机在这方面存在相当大的不同.

58

事实上，一个算法除了运算计数之外有更多其他方面的代价. 在单处理器计算机上，执行时间受到数据在存储层次的元件之间传送的影响，也受到在同一处理器上运行的竞争作业的影响. 在多处理器的机器上，情况变得更为复杂，有时处理器之间通信的重要性远远大于实际的"计算". 很遗憾，我们将忽略这些重要的考虑因素，因为本书本意是采用经典风格，集中在算法的基础上.

对于格拉姆-施密特迭代的两种变形，这里给出经典的结果.

定理8.1 计算一个 $m\times n$ 矩阵的QR因子分解，算法7.1和8.1需要 $\sim 2mn^2$ 次flop.

注意定理只表述了浮点运算计数的首项. 符号"~"有它通常的渐近意义：

$$\lim_{m,n\to\infty}\frac{\text{浮点运算数目}}{2mn^2}=1.$$

在算法的运算计数讨论中，标准的做法是舍弃低阶项，就像这里所做的那样，因为通常它们意义不大，除非 m 和 n 很小.

定理8.1可以如下证明. 为确定起见，考虑修正格拉姆-施密特算法，即算法8.1，当 m 和 n 很大时，工作量由最内部的循环支配：

$$r_{ij}=\boldsymbol{q}_i^*\boldsymbol{v}_j,$$
$$\boldsymbol{v}_j=\boldsymbol{v}_j-r_{ij}\boldsymbol{q}_i.$$

第一行计算内积 $\boldsymbol{q}_i^*\boldsymbol{v}_j$，需要 m 次乘法和 $m-1$ 次加法，第二行计算 $\boldsymbol{v}_j-r_{ij}\boldsymbol{q}_i$，需要 m 次乘法和 m 次减法. 在单个内迭代中全部工作量是 $\sim 4m$ 次flop，或每列向量元素4次flop. 全部加在一起，算法所需浮点运算次数渐近于

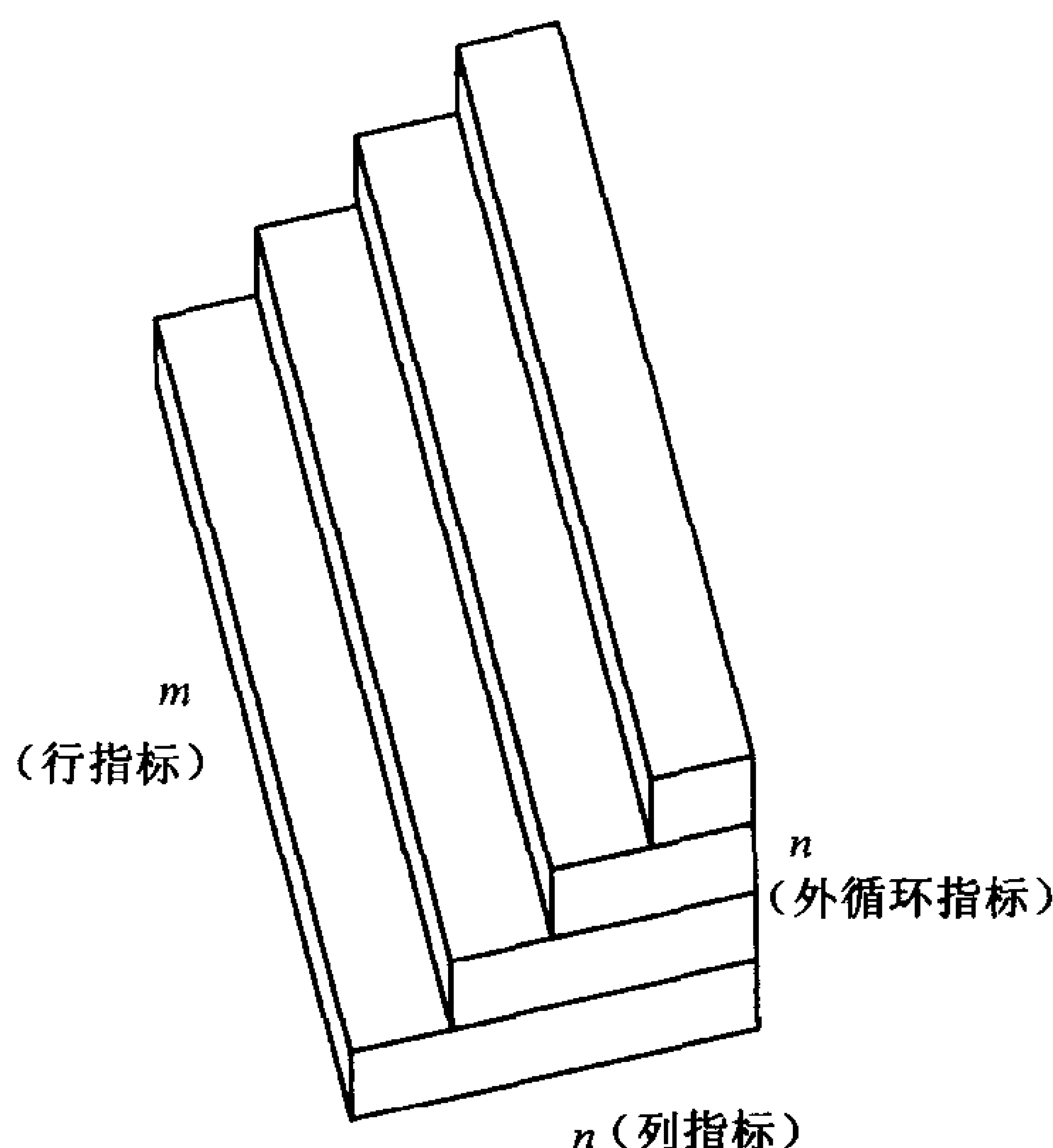
$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}4m\sim\sum_{i=1}^{n}(i)4m\sim 2mn^2. \tag{8.8}$$

8.4 用几何途径计算运算次数

运算计数总能像(8.8)那样由代数方法确定，这是数值分析文献中的标准方法，然而也可以开发一种不同的、几何途径来得到同样的结论. 论述如下，在外循

59

环的第 1 步，算法 8.1 在整个矩阵上运算，由其他各列减去第 1 列的一个倍数；在第 2 步，它在一个子矩阵上运算，由第 $3,\cdots,n$ 列减去第 2 列的一个倍数，按这种方式继续下去，每步列的数目缩减 1 个，直到最后一步，只有列 n 被修改. 此过程可用下图来表示：



底部的 $m\times n$ 矩形对应贯穿外循环的第 1 遍，其上面的 $m\times(n-1)$ 矩形对应第 2 遍，如此等等.

这样，当 $m,n\to\infty$ 时，格拉姆-施密特正交化运算计数的首项阶正比于上图的体积. 比例常数是 4 次 flop，因为如上所述，内循环的两步对应于每个矩阵位置的 4 次运算. 现当 $m,n\to\infty$ 时，图形收敛到一个直三角角柱，其体积为 $mn^2/2$. 每单位体积乘以 4 次 flop，再次给出

$$\text{格拉姆-施密特正交化的工作量为} \sim 2mn^2 \text{ 次 flop.} \tag{8.9}$$

本书一般用 (8.9) 的格式记录运算计数，而不将它们写成定理. 虽然代数推导也是可能的，但我们常常借助上述例子中那样的图来导出这些结果. 我们这样做的理由之一是，这种类型的图除了是运算计数的一条途径之外，也作为算法结构的一个提示. 对于不同结构的算法图形，参看第 75 和 154 页.

60

8.5　作为三角正交化的格拉姆-施密特

修正格拉姆-施密特算法的每个外层步可以解释为右乘一个上三角正方矩阵. 例如，由 $\boldsymbol{A}$ 开始，第 1 次迭代以 $1/r_{11}$ 乘列 $\boldsymbol{a}_1$，然后由每个剩余的列 $\boldsymbol{a}_j$ 减去 r_{1j} 再乘该结果，这等价于右乘矩阵 $\boldsymbol{R}_1$：

$$\begin{bmatrix} & & & \\ v_1 & v_2 & \cdots & v_n \\ & & & \end{bmatrix}\begin{bmatrix} \frac{1}{r_{11}} & \frac{-r_{12}}{r_{11}} & \frac{-r_{13}}{r_{11}} & \cdots \\ & 1 & & \\ & & 1 & \\ & & & \ddots \end{bmatrix} = \begin{bmatrix} & & & \\ q_1 & v_2^{(2)} & \cdots & v_n^{(2)} \\ & & & \end{bmatrix}.$$

一般地，算法 8.1 的第 i 步为当前 A 的列 $j(j>i)$ 减去 r_{ij}/r_{ii} 乘以列 i 得到新的矩阵的第 j 列，而用当前 A 的 i 列乘以 $1/r_{ii}$ 得到新的 i 列. 这对应于乘以一个上三角矩阵 R_i：

$$R_2 = \begin{bmatrix} 1 & & & \\ & \frac{1}{r_{22}} & \frac{-r_{23}}{r_{22}} & \cdots \\ & & 1 & \\ & & & \ddots \end{bmatrix}, \quad R_3 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \frac{1}{r_{33}} & \cdots \\ & & & \ddots \end{bmatrix}, \quad \cdots.$$

迭代的最后得

$$A \underbrace{R_1 R_2 \cdots R_n}_{\hat{R}^{-1}} = \hat{Q}. \tag{8.10}$$

此公式指出，格拉姆-施密特算法是一个三角正交化（triangular orthogonalization）的方法. 它把三角形的运算用在一个矩阵的右边，把它化为具有正交列的矩阵，当然，实际上我们并非显式地组成矩阵 R_i，并把它们乘在一起. 指出这些的目的是，使我们更深入地理解格拉姆-施密特算法的结构. 在第 20 讲我们将看到它与高斯消元法的结构十分相似.

习　题

8.1　令 A 为 $m \times n$ 矩阵，确定用算法 8.1 计算因子分解 $A=\hat{Q}\hat{R}$ 所含浮点加法、减法、乘法和除法的准确次数.

61

8.2　写出一个 MATLAB 函数 `[Q,R] = mgs(A)`（参看下一讲），它用修正格拉姆-施密特正交化计算 $m \times n$ 矩阵 A（$m \geqslant n$）的约化 QR 因子分解 $A=\hat{Q}\hat{R}$. 输出变量为具有正交列的矩阵 $Q \in \mathbb{C}^{m\times n}$ 及三角矩阵 $R \in \mathbb{C}^{n\times n}$.

8.3　本页的每个上三角矩阵 R_j 可以解释为，一个对角矩阵和一个单位上三角矩阵（即一个以 1 为对角线元素的上三角矩阵）的乘积. 准确地解释这些因子，并且解释算法 8.1 的哪一行对应每个因子.

62

第 9 讲　MATLAB

学习数值线性代数，一定要养成在计算机上做实验的习惯. 要做到这一点，没有比使用名为 MATLAB® 的问题求解环境更好的方法了. 在本讲中用 3 个例子来说明MATLAB实验，沿此途径，我们得到关于格拉姆-施密特正交化稳定性的一些结论.

MATLAB 是一种以向量和矩阵作为基本数据类型的数学计算语言. 它和 Fortran 和 C 那样的语言区别在于，它在较高的数学水平上运算，包括作为内部命令的矩阵求逆、奇异值分解、快速傅里叶变换等数百种运算. 它也是一个问题求解环境，用一个解释程序而非编译程序产生顶层注释，而且提供直接对 2D 和 3D 图像的访问.

自 20 世纪 80 年代以来，MATLAB 已经成为全世界数值分析学家和工程师广泛应用的工具. 对于很多大规模科学计算问题，以及对于现实所有中小规模的数值线性代数实验，它都是可选的语言.

在本书中，将使用 MATLAB 并给出某些数值实验，在一些习题中也这样. 我们并不系统地描述这种语言，因为我们给出的实验的数目是有限的，而且只需要有阅读 MATLAB 的能力就可以了.

实验 1：离散勒让德多项式

在第 7 讲考虑了范德蒙德“矩阵”，其“列”包含了在区间 [-1,1] 上的单项式 $1,x,x^2$ 和 x^3. 现在假设用 257 个等距离的点将 [-1,1] 离散，产生一个真正的范德蒙德矩阵. 下面的 MATLAB 行构造了这个矩阵，并计算它的约化 QR 因子分解.

```
x = ( -128:128)'/128;             置 x 为[ -1,1]的离散化.
A = [x.^0 x.^1 x.^2 x.^3];        构造范德蒙德矩阵.
[Q,R] = qr(A,0);                  求它的 QR 因子分解.
```

在这，对这些命令需要说明几点. 在第 1 行，撇号'将 (-128: 128) 由行转为列向量. 在第 2 行，系列的.^指各项的 (entrywise) 的幂. 在第 3 行，qr 是计算 QR 因子分解的内部 MATLAB 函数，并指出约化而非完全因子分解的变元 0 是必需的. 这里用的方法不是格拉姆-施密特正交化而是豪斯霍尔德三角形化，这将在下一讲中讨论，但它不是为了当前目的所得的结果. 在这 3 行中，如末尾的分号删去，则打印

输出(x,A,Q,和 R).

矩阵 Q 的列本质上是图7-1中前4个勒让德多项式，它们有点不同，总的来说接近图的精确度，因为在定义了勒让德多项式的 $[-1,1]$ 上的连续内积已被一个离散模拟代替了．它们在正规化方面也有所不同，因为勒让德多项式应该满足 $P_k(1)=1$，我们可以用 Q 的每列最后的元素除该列来确定．下面的 MATLAB 行用右乘一个 4×4 对角矩阵做到这一点．

```
scale = Q(257,:);                 选 Q 最后的行.
Q = Q * diag(1./scale);           用这些数改变列的比例.
plot(Q)                           作重定比例后 Q 的列的图.
```

计算的结果是像图7-1那样的曲线（不显示）．在Fortran或C中，这会写成包含大量的循环和嵌套循环的很多行程序．在我们的 MATLAB 6行程序中，没有明显地单独出现一个循环，虽然在每行中至少隐含有一个循环．

64

实验2：经典和修正格拉姆-施密特算法的对比

第2个例子有更多算法的内容，它的目的是测试经典和修正格拉姆-施密特算法之间数值稳定性的区别．

首先，构造一个方阵 A，它有随机的奇异向量且有变化范围极大的奇异值，奇异值的范围是 2^{-1} 到 2^{-80} 之间2的因子．

```
[U,X] = qr(randn(80));            置 U 为随机正交矩阵.
[V,X] = qr(randn(80));            置 V 为随机正交矩阵.
S = diag(2.^(-1:-1:-80));         置 S 为有指数分级元素的对角矩阵.
A = U * S * V;                    置 A 为以这些元素作为奇异值的矩阵.
```

现在用算法7.1和8.1计算 A 的QR因子分解，在下面的代码中，clgs 和 mgs 是算法7.1和8.1的实现，在此不列出．

```
[QC,RC] = clgs(A);        用经典格拉姆-施密特计算因子分解 Q^(c) R^(c).
[QM,RM] = mgs(A);         用修正格拉姆-施密特计算因子分解 Q^(m) R^(m).
```

最后，我们做出由这两种计算产生的对角元素 r_{jj} 的图（对应的 MATLAB 代码未显示）．因为 $r_{jj}=\|P_j a_j\|$，这给出了在每步投影大小的图形，这个结果在图9-1中以对数尺度显示．

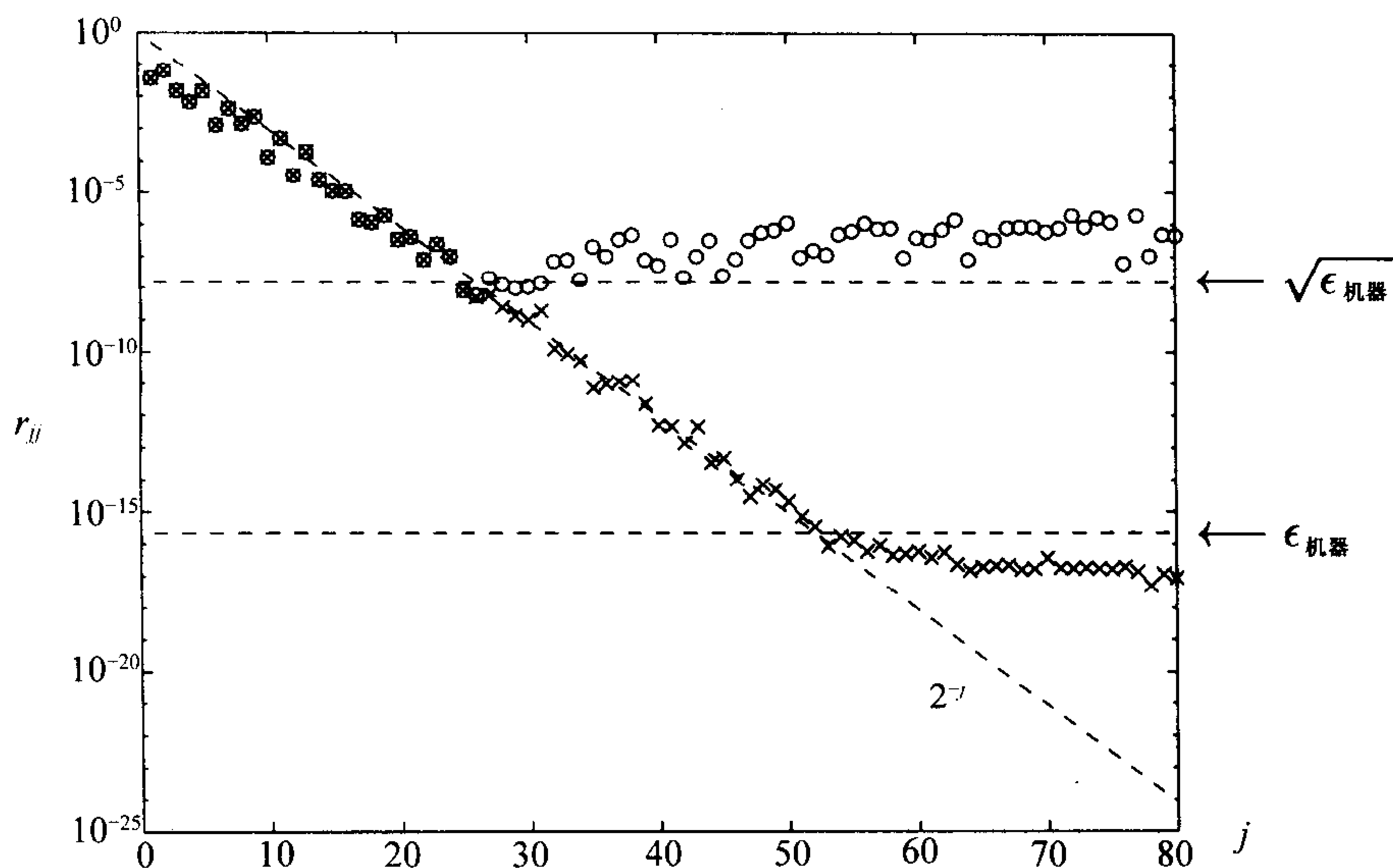


图 9-1 对一个指数分级奇异值矩阵的 QR 因子分解，计算的 r_{jj} 作为 j 的函数. 在此计算机约有 16 位十进制数字的相对准确度，经典格拉姆–施密特算法产生的数用圆圈表示，而修正格拉姆–施密特算法产生的数用叉号表示

在图中，首先注意的是 r_{jj} 对 j 有一个稳定的下降，接近符合线 2^{-j}. 显然，r_{jj} 不精确等于 A 的第 j 个奇异值，但它是一个相当好的近似. 这个现象可以粗略地作如下解释，A 的 SVD 可用（5.3）的形式写成

$$A = 2^{-1}u_1v_1^* + 2^{-2}u_2v_2^* + 2^{-3}u_3v_3^* + \cdots + 2^{-80}u_{80}v_{80}^*,$$

其中，$\{u_j\}$ 和 $\{v_j\}$ 分别是 A 的左和右奇异向量，特别地，A 的第 j 列有形式

$$a_j = 2^{-1}\bar{v}_{j1}u_1 + 2^{-2}\bar{v}_{j2}u_2 + 2^{-3}\bar{v}_{j3}u_3 + \cdots + 2^{-80}\bar{v}_{j,80}u_{80}.$$

因奇异向量是随机的，我们可以期望数 $\bar{v}_{ji}$ 都有相似的，其阶为 $80^{-1/2} \approx 0.1$ 的数量. 现在，当我们作 QR 因子分解时，显然第 1 个向量 q_1 大概近似等于 u_1，同时有 $2^{-1} \times 80^{-1/2}$ 阶的 r_{11}. 下一步的正交化将产生近似等于 u_2 的第 2 个向量 q_2，同时有 $2^{-2} \times 80^{-1/2}$ 阶的 r_{22}，如此等等.

65

在图 9-1 中，注意的第 2 件事是 r_{jj} 的几何下降并非一路继续到 $j = 80$，这是计算机的舍入误差所致. 对经典格拉姆–施密特算法，这些数永远不会变得小于 10^{-8} 左右. 对修正格拉姆–施密特算法，它们将缩小 8 个数量阶，跌至 10^{-16} 阶，这是用于此计算的计算机的机器 ϵ（machine epsilon）水平，机器 ϵ 将在第 13 讲中定义.

显然，某些算法要比其他算法更稳定，由上可见，经典格拉姆–施密特过程是不

稳定的算法之一，因此它罕有应用，除了有时在并行计算机上，这时关于通信的优点比不稳定的缺点更加重要.

实验3：正交性的数值损失

冒着连续给出两个不稳定现象来困扰读者的风险，我们展示另外一个例子来结束本讲，这是同时影响经典和修正格拉姆-施密特算法两者的另一种不同的不稳定性类型. 在浮点运算中，这些算法可以产生远非正交的向量 $\boldsymbol{q}_j$. 正交性的损失出现在当 $\boldsymbol{A}$ 接近秩亏损时，也像大多数的不稳定性，它甚至可以出现在低维情形.

66

不用 MATLAB，我们在纸上考虑矩阵

$$\boldsymbol{A}=\begin{bmatrix}0.70000 & 0.70711\\ 0.70001 & 0.70711\end{bmatrix} \tag{9.1}$$

在一个将所有计算结果舍入为相对准确的5位十进制数字（第13讲）的计算机上的情形. 在 2×2 情形下，经典和修正算法是等同的. 在步 $j=1$ 上，第一列被正规化，在5位数字算术中产生

$$r_{11}=0.98996,\quad \boldsymbol{q}_1=\boldsymbol{a}_1/r_{11}=\begin{bmatrix}0.70000/0.98996\\ 0.70001/0.98996\end{bmatrix}=\begin{bmatrix}0.70710\\ 0.70711\end{bmatrix}.$$

在步 $j=2$，计算出 $\boldsymbol{a}_2$ 在 $\boldsymbol{q}_1$ 方向的分量，并减去：

$$r_{12}=\boldsymbol{q}_1^*\boldsymbol{a}_2=0.70710\times0.70711+0.70711\times0.70711=1.0000,$$

$$v_2=\boldsymbol{a}_2-r_{12}\boldsymbol{q}_1=\begin{bmatrix}0.70711\\ 0.70711\end{bmatrix}-\begin{bmatrix}0.70710\\ 0.70711\end{bmatrix}=\begin{bmatrix}0.00001\\ 0.00000\end{bmatrix},$$

这里再次用5位数字舍入的结果，这个计算 v_2 的结果由误差主导. 最后计算出的 $\boldsymbol{Q}$ 是

$$\boldsymbol{Q}=\begin{bmatrix}0.70710 & 1.0000\\ 0.70711 & 0.0000\end{bmatrix},$$

它并不接近任何正交矩阵.

在一个有16位数字精度的计算机上，若把修正格拉姆-施密特应用于矩阵(9.1)，仍然要损失正交性的约5位数字. 这里是 MATLAB 的数据资料，“eye”函数产生已指出维数的单位矩阵.

$A = \begin{bmatrix} 0.70000 & .70711 \\ 0.70001 & .70711 \end{bmatrix}$;	定义 A.
`[Q,R] = qr(A);`	由豪斯霍尔德计算因子 Q.
`norm(Q'*Q - eye(2))`	判别 Q 的正交性.
`[Q,R] = mgs(A);`	由修正 G-S 计算因子 Q.
`norm(Q'*Q - eye(2))`	判别 Q 的正交性.

没有分号的行产生下列打印结果:

```
ans = 2.3515e - 16,     ans = 2.3014e - 11.
```

67

习 题

9.1 (a) 运行实验 1 的 6 行 MATLAB 程序,产生一个近似勒让德多项式的图.

(b) 对 $k=0,1,2,3$,作这些近似多项式与准确多项式 (7.11) 在 257 个格子点上之差的图. 误差有多大? 它们如何分布?

(c) 对于格子距离 $\Delta x = 2^{-v}$ 取其他的 v 值时,比较这些结果. Δx 的什么幂出现来控制收敛性?

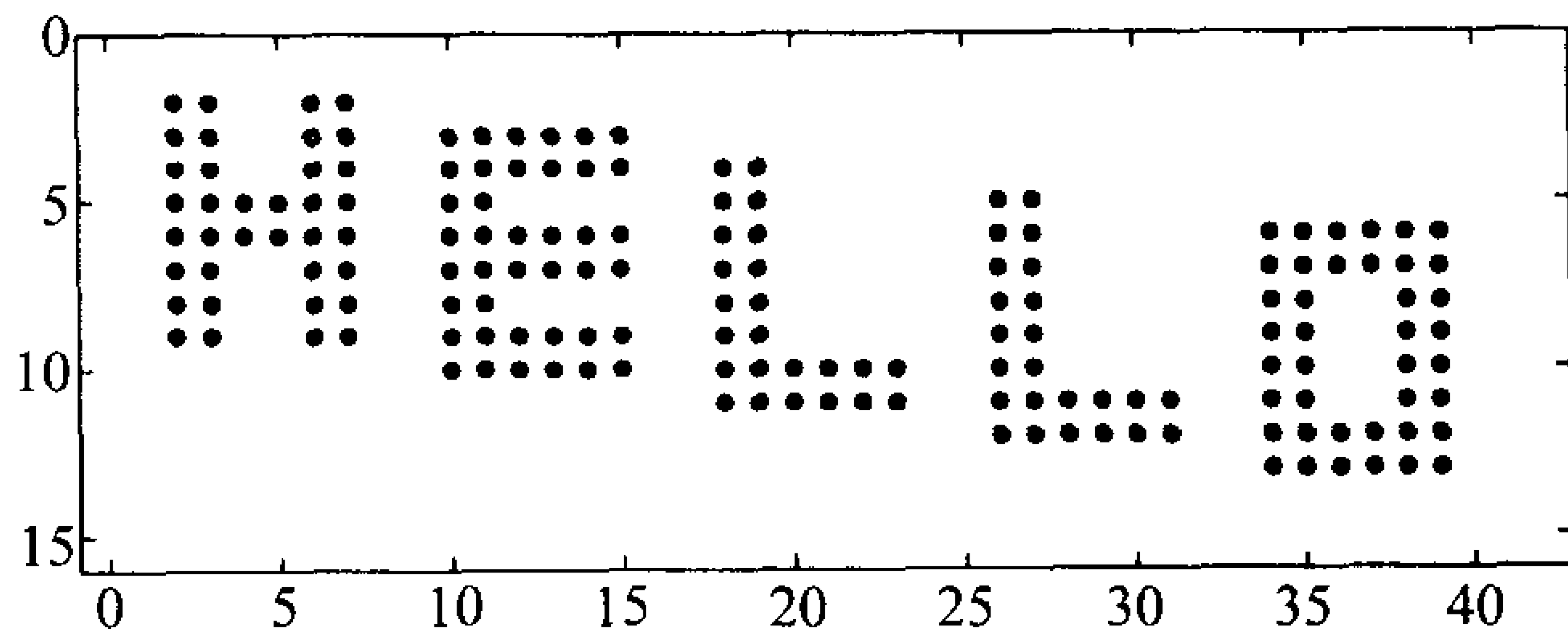
9.2 在实验 2 中, A 的奇异值近似地与 QR 因子 R 的对角元相配. 现在考虑一个非常不同的例子,假设 $Q=I$, $A=R$ 是主对角线上为 1,第 1 条次对角线上为 2 且其他处均为 0 的 $m \times m$ 矩阵 (Toeplitz 矩阵).

(a) A 的特征值,行列式和秩是什么?

(b) A^{-1} 是什么矩阵?

(c) 给出 A 的第 m 个奇异值 σ_m 的一个非平凡上界. 为了鼓励大家,欢迎使用 MATLAB,但你给出的界应该能够分析地证明. (提示: 用 (b).)

9.3 (a) 写出一个 MATLAB 程序,用于建立除了在下图中指定位置的元素为 1 外,其余各处元素均为 0 的 15 × 40 矩阵. 最左上角的 1 位于 (2,2),最右下角的 1 位于 (13,39),这个图像由命令 spy(A) 产生.



(b) 调用 svd 计算 A 的奇异值,并打印其结果. 用 plot 和 semilogy 做出这些数的图. A 的秩在数学上准确地等于什么? 在计算的奇异值中怎样表现出来?

(c) 对于由 1 到 rank(A) 的每个 i,构造秩 i 的矩阵 B,它是 2-范数意义下 A 的最佳逼近. 用命令 pcolor(B) 及 colormap(gray) 形成这些各种逼近的像.

68

第 10 讲　豪斯霍尔德三角形化

计算 QR 因子分解的另一个主要方法是豪斯霍尔德三角形化，它比格拉姆-施密特正交化更加稳定，尽管它缺少后者具有的作为迭代法基础的适应性. 豪斯霍尔德算法是一个“正交三角形化”的过程，以系列的酉矩阵运算形成一个三角矩阵.

10.1　豪斯霍尔德和格拉姆-施密特

如在第 8 讲中所见，格拉姆-施密特迭代在 A 的右边逐次应用初等三角矩阵 R_k，所得矩阵

$$A\underbrace{R_1R_2\cdots R_n}_{\hat{R}^{-1}} = \hat{Q}$$

有正交的列. 乘积 $\hat{R} = R_n^{-1}\cdots R_2^{-1}R_1^{-1}$ 也是上三角的，因此 $A = \hat{Q}\hat{R}$ 是 A 的一个约化 QR 因子分解.

69

与之对照，豪斯霍尔德方法在 A 的左边逐次应用初等酉矩阵 Q_k，所得矩阵

$$\underbrace{Q_n\cdots Q_2Q_1}_{Q^*}A = R$$

是上三角的. 乘积 $Q = Q_1^*Q_2^*\cdots Q_n^*$ 也是酉矩阵，因此 $A = QR$ 是 A 的一个完全 QR 因子分解.

这两个方法可以总结为：

格拉姆-施密特：三角形正交化.

豪斯霍尔德：正交三角形化.

10.2　以引入零元素实现三角形化

豪斯霍尔德方法的核心是原来由 A. 豪斯霍尔德在 1958 年提出的一个思想. 这是设计酉矩阵 Q_k，使得 $Q_n\cdots Q_2Q_1A$ 为上三角阵的一个别出心裁的方法.

选择矩阵 Q_k，使得在第 k 列对角以下引入零元素，而保持先前引入的零元素不变. 例如，在 5×3 情形，3 次用到的运算 Q_k 如下. 在这些矩阵中符号 $\times$ 表示不必为零的元素，黑粗体 $\boldsymbol{\times}$ 表示刚刚改变了的元素，空白元素为零.

$$
\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}
\xrightarrow{Q_1}
\begin{bmatrix} \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} \end{bmatrix}
\xrightarrow{Q_2}
\begin{bmatrix} \times & \times & \times \\ & \mathbf{\times} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \\ & \mathbf{0} & \mathbf{\times} \end{bmatrix}
\xrightarrow{Q_3}
\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & \mathbf{\times} \\ & & \mathbf{0} \\ & & \mathbf{0} \end{bmatrix}
\tag{10.1}
$$

$$A \qquad\qquad Q_1A \qquad\qquad Q_2Q_1A \qquad\qquad Q_3Q_2Q_1A$$

首先，Q_1 在第 $1,\cdots,5$ 行上运算，在位置 $(2,1)$，$(3,1)$，$(4,1)$ 和 $(5,1)$ 引入零. 然后，Q_2 在第 $2,\cdots,5$ 行上运算，在位置 $(3,2)$，$(4,2)$ 和 $(5,2)$ 引入零而不破坏由 Q_1 引入的零. 最后，Q_3 在第 $3,\cdots,5$ 行上运算，在位置 $(4,3)$ 和 $(5,3)$ 引入零而不破坏早先引入的任何零元素.

一般地，Q_k 在第 $k,\cdots,m$ 行上运算. 在第 k 步的开始，这些行的前 $k-1$ 列有一个零元素的块，Q_k 的应用组成了这些行的线性组合，而零元素的线性组合仍为零. n 步之后，所有对角线下的元素已经消去，$Q_n\cdots Q_2Q_1A=R$ 为上三角阵.

10.3　豪斯霍尔德镜射算子

怎样构造酉矩阵 Q_k 以引入（10.1）中指出的零？标准的方法如下，每个 Q_k 选为一个形如

$$Q_k=\begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}, \tag{10.2}$$

的酉矩阵，其中 I 为 $(k-1)\times(k-1)$ 单位矩阵，F 为一个 $(m-k+1)\times(m-k+1)$ 酉矩阵，乘以 F 一定要将零引入第 k 列. 豪斯霍尔德算法选择 F 为一个特殊的矩阵，它称为*豪斯霍尔德镜射算子*（Householder reflector）. [70]

假设在第 k 步的开始，第 k 列的第 $k,\cdots,m$ 个元素由向量 $x\in\mathbb{C}^{m-k+1}$ 给出. 为把修改的零引入到第 k 列，豪斯霍尔德镜射算子 F 应有下述映射的效果：

$$x=\begin{bmatrix} \times \\ \times \\ \times \\ \vdots \\ \times \end{bmatrix} \xrightarrow{F} Fx=\begin{bmatrix} \|x\| \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}=\|x\|e_1. \tag{10.3}$$

（我们将马上以 $\pm$ 号修改这个想法.）实现思想如图 10-1 所示. 镜射算子 F 将横跨正交于 $v=\|x\|e_1-x$ 的超平面 H 来镜射空间 $\mathbb{C}^{m-k+1}$. *超平面*（hyperplane）是三维空间中二维平面的推广——四维空间中的三维子空间，五维空间中的四维子空间等等. 一般地，一个超平面可以用正交于某一固定非零向量为其特征. 在图 10-1 中这个向

量是 $v = \|x\|e_1 - x$，可以想像虚线是把 H 描绘成“切口”.

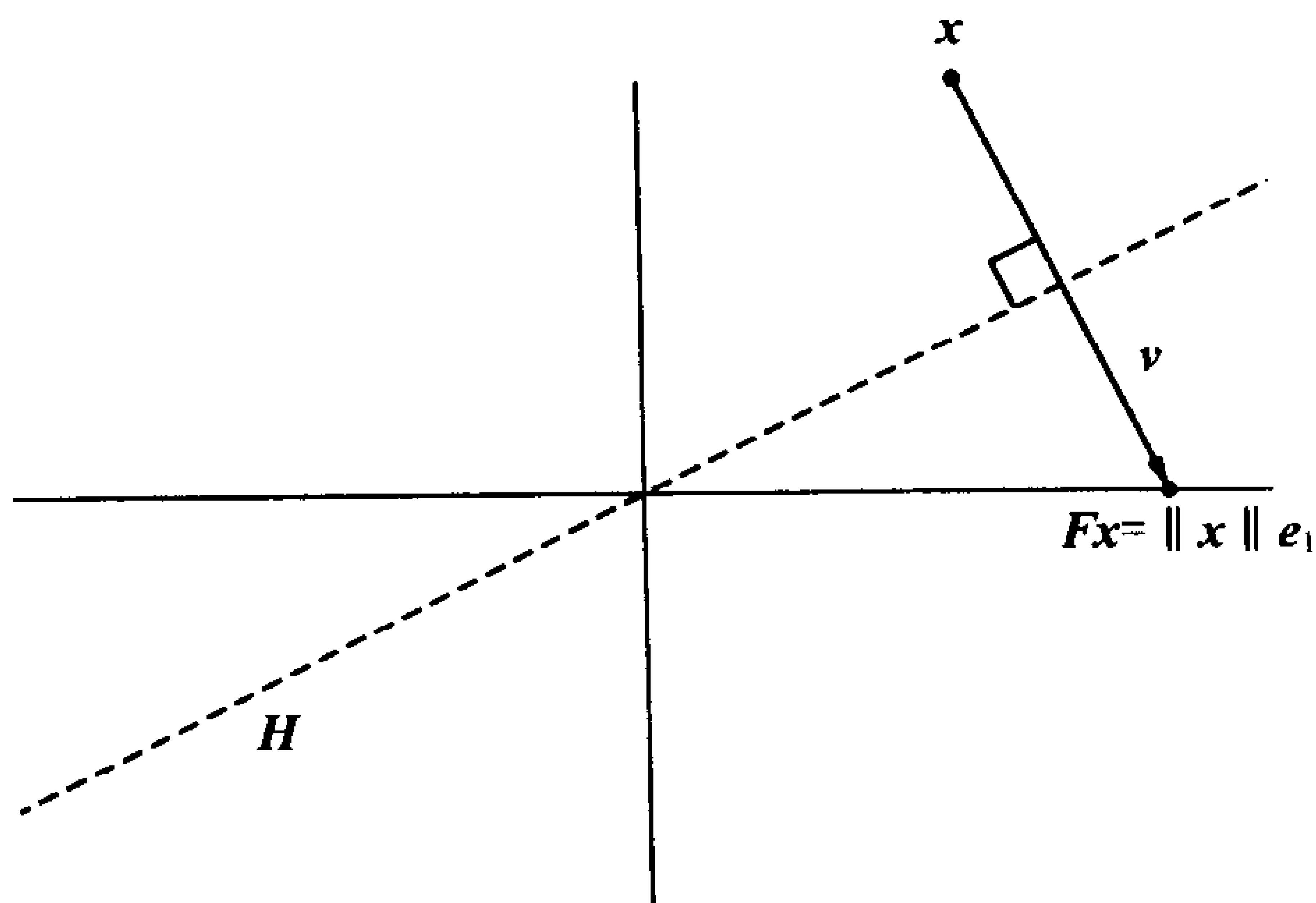


图 10-1　豪斯霍尔德镜射算子

当应用镜射算子时，把位于超平面 H 一边的每个点映射到它在 H 另一边的镜像. 具体而言就是把 x 映射到 $\|x\|e_1$. 这个镜射的公式推导如下. 在（6.11）曾见到对任意的 $y \in \mathbb{C}^m$，向量

71

$$Py = \left(I - \frac{vv^*}{v^*v}\right)y = y - v\left(\frac{v^*y}{v^*v}\right)$$

是 y 在空间 H 上的正交投影. 为了横跨 H 镜射 y，一定不能停在这个点上，而要在点到 H 方向上准确地走到点离 H 平面距离的 2 倍远处. 因此，镜射 Fy 应为

$$Fy = \left(I - 2\frac{vv^*}{v^*v}\right)y = y - 2v\left(\frac{v^*y}{v^*v}\right).$$

这样，矩阵 F 是

$$F = I - 2\frac{vv^*}{v^*v}. \tag{10.4}$$

注意，投影算子 P（秩 $m-1$）与镜射算子 F（满秩，酉）仅在因子 2 的引入上不同.

10.4　两个镜射算子的比较

在（10.3）和图 10-1 中已经做了简化，因为事实上存在许多豪斯霍尔德镜射可以引入所需的零元素. 向量 x 可以镜射至 $z\|x\|e_1$，其中 z 为满足 $|z|=1$ 的任意数量. 在复数情形，存在可能镜射的一个圆，而在实数情形，存在两种可供的选择，它由

横跨两个不同的超平面 $\boldsymbol{H}^-$ 和 $\boldsymbol{H}^+$ 的镜射表示，如在图 10-2 所作的说明.

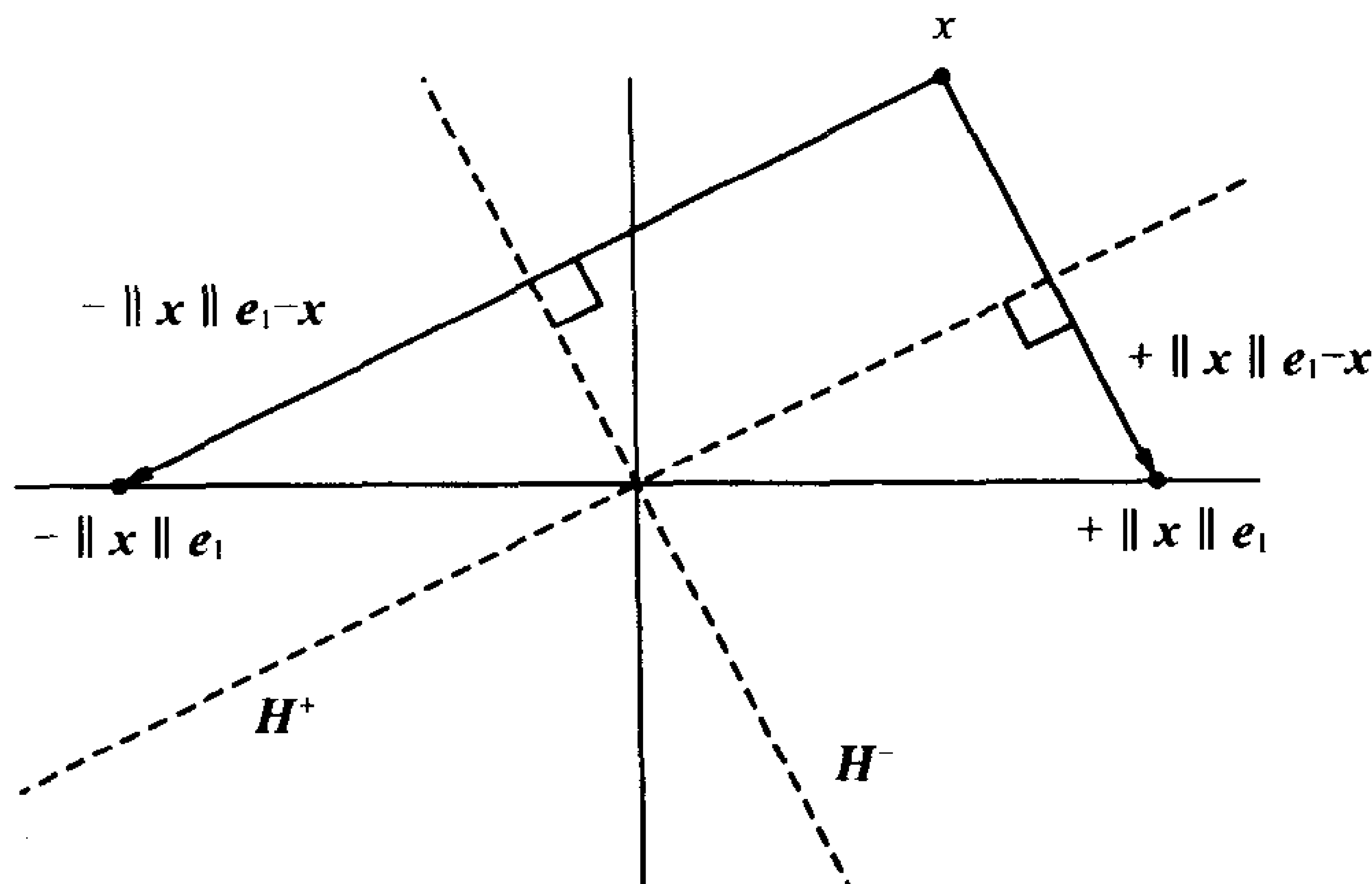


图 10-2　两种可能的镜射. 为了数值的稳定性，重要的是选择最大距离移动 $\boldsymbol{x}$ 的那一种

从数学上看，符号的每种选择都令人满意. 然而，为了数值稳定性的目的——对舍入误差的不灵敏——要求限定一种选择而不考虑其他. 为了数值稳定性，希望镜射 $\boldsymbol{x}$ 到不太接近 $\boldsymbol{x}$ 自身的向量 $z\|\boldsymbol{x}\|\boldsymbol{e}_1$. 为了做到这一点，可以选择 $z=-\mathrm{sign}(x_1)$，其中 x_1 记为 $\boldsymbol{x}$ 的第一个分量，这样，镜射向量成为 $\boldsymbol{v}=-\mathrm{sign}(x_1)\|\boldsymbol{x}\|\boldsymbol{e}_1-\boldsymbol{x}$，或者去掉因子 -1， [72]

$$\boldsymbol{v}=\mathrm{sign}(x_1)\|\boldsymbol{x}\|\boldsymbol{e}_1+\boldsymbol{x}. \tag{10.5}$$

为了使这成为一个完备的规定，可以随意地加上限制：当 $x_1=0$ 时，$\mathrm{sign}(x_1)=1$.

不难看到，为什么符号的选择会造成稳定性的不同. 假设在图 10-2 中，$\boldsymbol{H}^+$ 和 $\boldsymbol{e}_1$ 轴之间的角度非常小，这样向量 $\boldsymbol{v}=\|\boldsymbol{x}\|\boldsymbol{e}_1-\boldsymbol{x}$ 比 $\boldsymbol{x}$ 或 $\|\boldsymbol{x}\|\boldsymbol{e}_1$ 小很多. 因而 $\boldsymbol{v}$ 的计算就是相近量的减法，而将受到消去误差的影响. 如果像（10.5）那样取符号，则保证了 $\|\boldsymbol{v}\|$ 永不小于 $\|\boldsymbol{x}\|$，就避免了这种影响.

10.5　算　　法

现在给出全部豪斯霍尔德算法的公式，为此，用一个新的（MATLAB 方式的）记号更方便. 若 $\boldsymbol{A}$ 是一个矩阵，定义 $\boldsymbol{A}_{i:i',j:j'}$ 是 $\boldsymbol{A}$ 的一个 $(i'-i+1)\times(j'-j+1)$ 子矩阵，其左上角为 a_{ij}，右下角为 $a_{i',j'}$. 当子矩阵化为单行或单列子向量的特殊情形时，就分别写成 $\boldsymbol{A}_{i,j:j'}$ 或 $\boldsymbol{A}_{i:i',j}$.

下列算法计算一个 $m\times n$（$m\geqslant n$）矩阵 $\boldsymbol{A}$ 的 QR 因子分解的因子 $\boldsymbol{R}$，占用以前 $\boldsymbol{A}$ 的存储空间. 在这个过程中，保存了 n 个镜射向量 $\boldsymbol{v}_1,\cdots,\boldsymbol{v}_n$，以备后用.

算法 10.1　豪斯霍尔德 QR 因子分解

for $k=1$ to n

　　$x=A_{k:m,k}$

　　$v_k=\text{sign}(x_1)\|x\|_2e_1+x$

　　$v_k=v_k/\|v_k\|_2$

　　$A_{k:m,k:n}=A_{k:m,k:n}-2v_k(v_k^*A_{k:m,k:n})$

10.6　应用或形成 Q

在算法 10.1 结束时，A 已经化为上三角形式，这就是 QR 因子分解 $A=QR$ 中的矩阵 R. 然而酉矩阵 Q 尚未被构造出来，也没有它对应约化 QR 因子分解的 n 列子矩阵$\hat{Q}$. 究其原因是因为构造 Q 或$\hat{Q}$要有额外的工作，而且在很多应用中，我们也可以避免这样做，这只要通过直接地处理公式

$$Q^* = Q_n\cdots Q_2Q_1 \tag{10.6}$$

或其共轭

73

$$Q = Q_1Q_2\cdots Q_n. \tag{10.7}$$

即可.（这里并非忘记了星号，而是因为这里的每个 Q_j 都是埃米尔特矩阵.）

例如，在第 7 讲中见到，正方的方程组 $Ax=b$ 可以借助 A 的 QR 因子分解求解. 在此过程中，惟一要用到 Q 的是乘积 Q^*b 的计算. 由（10.6）可知，将应用于 A 上，使 A 成为三角阵的那 n 次运算运用在 b 上即可得到 Q^*b. 算法如下.

算法 10.2　乘积 Q^*b 的隐式计算

for $k=1$ to n

　　$b_{k:m}=b_{k:m}-2v_k(v_k^*b_{k:m})$

类似地，乘积 Qx 的计算可以用同样的过程以相反的次序来完成.

算法 10.3　乘积 Qx 的隐式计算

for $k=n$ down to 1

　　$x_{k:m}=x_{k:m}-2v_k(v_k^*x_{k:m})$

这两个算法的每个所含的工作量是 $O(mn)$阶，而不是算法 10.1 的 $O(mn^2)$阶（见下述）.

当然，有时会希望显式地构造矩阵 Q，这可用各种方法来完成. 可以借助算法

10.3 计算 $\boldsymbol{QI}$ 的列 $\boldsymbol{Qe}_1,\boldsymbol{Qe}_2,\cdots,\boldsymbol{Qe}_m$ 来构造 $\boldsymbol{QI}$，也可借助算法 10.2 先构造 $\boldsymbol{Q}^*\boldsymbol{I}$，然后对其结果取共轭. 这种想法也可稍作改变，就是在每一步而非对最后的乘积取共轭，即：由（10.7）所启发的那样，通过计算 $\boldsymbol{IQ}$ 的行 $e_1^*\boldsymbol{Q},e_2^*\boldsymbol{Q},\cdots,e_m^*\boldsymbol{Q}$ 来构造 $\boldsymbol{IQ}$. 对于这几种想法，最好的是基于算法 10.3 的第一种. 理由是它从 $\boldsymbol{Q}_n,\boldsymbol{Q}_{n-1}$ 等的运算开始，只修改被用到的向量的一小部分. 如果利用了这个稀疏性质的优点，计算速度将会加快.

如果只需 $\hat{\boldsymbol{Q}}$ 而非 $\boldsymbol{Q}$，则计算列 $\boldsymbol{Qe}_1,\boldsymbol{Qe}_2,\cdots,\boldsymbol{Qe}_n$ 就足够了.

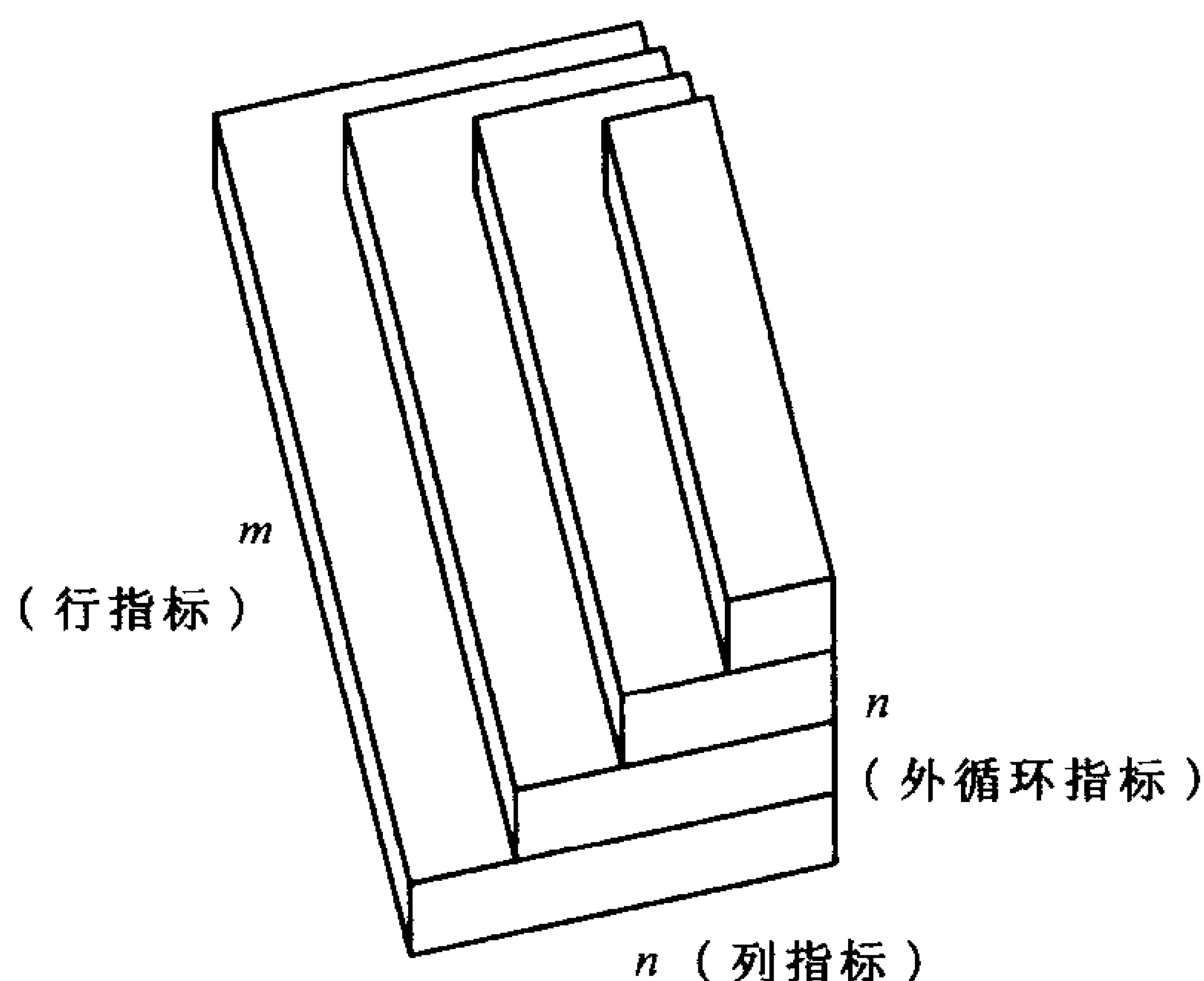
10.7 运算计数

算法 10.1 所含的工作量由最内层循环

$$A_{k:m,j}-2v_k(v_k^*A_{k:m,j}). \tag{10.8}$$

支配. 如果向量长度是 $l=m-k+1$，那么这个计算需要 $4l-1\sim 4l$ 次数量运算：l 次减法，l 次数量乘法和 $2l-1$ 次点积. 对每个在其上运算的元素，这是 ~4 次 flop. 74

如同在第 8 讲中，可以用几何方法把这些每个元素的 4 次 flop 加起来. 外循环的每步运算只针对较少的行，因为在第 k 步当中，第 $1,\cdots,k-1$ 行是不变的. 进而，每步也只在较少的列上运算，因为参与运算的这些行的第 $1,\cdots,k-1$ 列为零而被忽略过去. 因此一个外循环步的工作量可表示为下列几何体的一个单层：



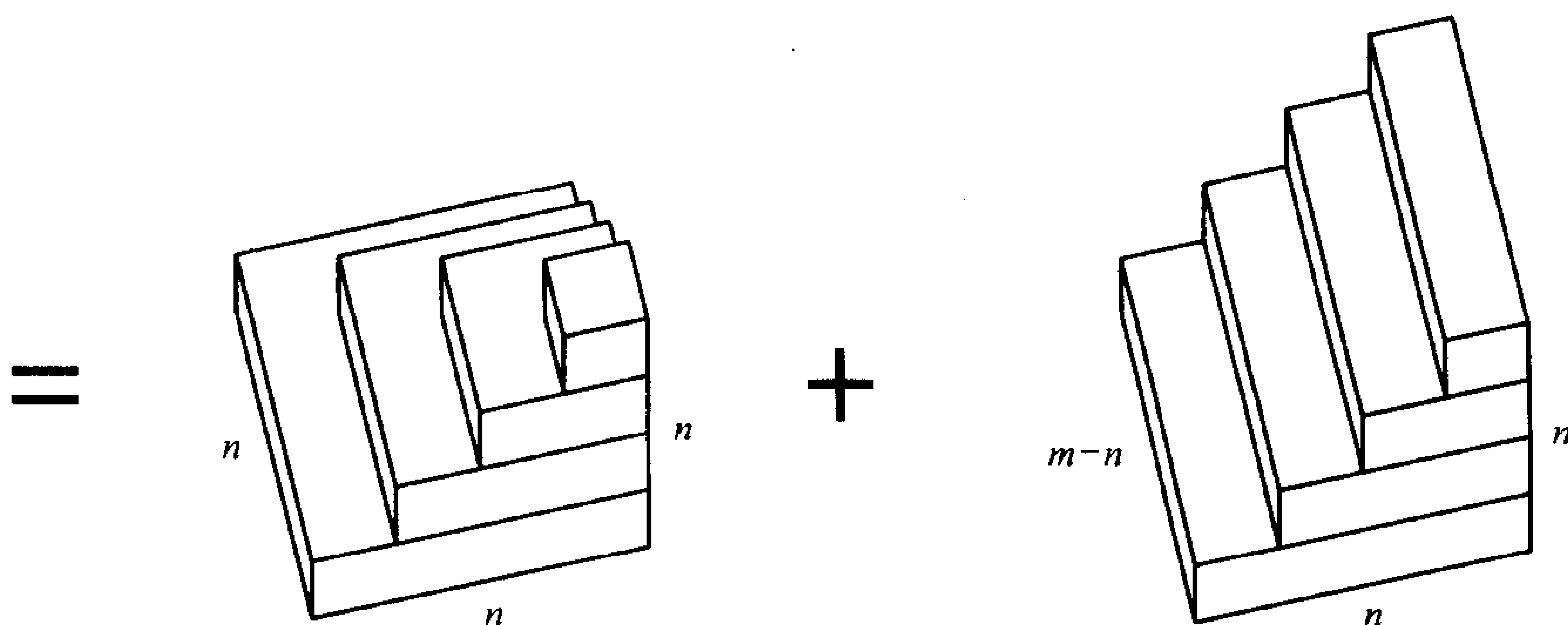
全部运算的数目对应于 4 倍这个几何体的体积. 为了从图上确定这个体积，可将此几何体分为两块（见下页图）.

左边的几何体呈古代巴比伦庙塔的形状，当 $n\to\infty$ 时，收敛到一个角锥体，其体积为 $\frac{1}{3}n^3$. 右边的几何体呈楼梯状，当 $m,n\to\infty$ 时，收敛到一个棱柱，其体积为

$\frac{1}{2}(m-n)n^2$. 加起来，其体积为 $\sim\frac{1}{2}mn^2-\frac{1}{6}n^3$. 每单位体积乘以4次flop，得到

75

$$\text{豪斯霍尔德正交化的工作量：}\sim 2mn^2-\frac{2}{3}n^3\text{ 次 flop.} \tag{10.9}$$



习　题

10.1　确定豪斯霍尔德镜射算子的（a）特征值，（b）行列式和（c）奇异值，对特征值，除了代数证明之外给出一个几何论证.

10.2　（a）写出一个 MATLAB 函数`[W,R] = house(A)`，它用豪斯霍尔德镜射计算一个 $m\times n$ 矩阵 $\boldsymbol{A}$（$m\geqslant n$）的完全 QR 因子分解 $\boldsymbol{A}=\boldsymbol{QR}$ 的隐式表示. 输出变量为一个其列是逐次豪斯霍尔德镜射确定的向量 $\boldsymbol{v}_k$ 的下三角矩阵 $\boldsymbol{W}\in\mathbb{C}^{m\times n}$，及三角矩阵 $\boldsymbol{R}\in\mathbb{C}^{n\times n}$.

（b）写出一个 MATLAB 函数 `Q = formQ(W)`，它以由 house 产生的矩阵 W 作为输入，产生对应的 $m\times m$ 正交矩阵 $\boldsymbol{Q}$.

10.3　令 $\boldsymbol{Z}$ 为矩阵

$$\boldsymbol{Z}=\begin{bmatrix}1&2&3\\4&5&6\\7&8&7\\4&2&3\\4&2&2\end{bmatrix}.$$

在 MATLAB 上计算 $\boldsymbol{Z}$ 的三种约化 QR 因子分解：用习题8.2的格拉姆-施密特程序 mgs，用习题10.2的豪斯霍尔德程序 house 及用 MATLAB 的内置命令`[Q,R] = qr(Z,0)`. 比较三者并评论你所看到的任何不同之处.

10.4　考虑 2×2 正交矩阵

$$\boldsymbol{F}=\begin{bmatrix}-c&s\\s&c\end{bmatrix},\quad \boldsymbol{J}=\begin{bmatrix}c&s\\-s&c\end{bmatrix}, \tag{10.10}$$

其中对某个 θ，$s=\sin\theta$ 及 $c=\cos\theta$，第一个矩阵有 $\det F=-1$ 且它是一个镜射算子——豪斯霍尔德算子在二维的特殊情形，第二个矩阵有 $\det J=1$，其效果是旋转而非镜射. 这样的矩阵称为吉文斯旋转.

(a) 准确地描述在平面 $\mathbb{R}^2$ 上左乘 $\boldsymbol{F}$ 及 $\boldsymbol{J}$ 的几何效果.（例如，$\boldsymbol{J}$ 表示将平面旋转角度 θ，但旋转是顺时针还是逆时针的?）

(b) 描述一个 QR 因子分解算法，它类似于算法 10.1 但是基于吉文斯旋转而非豪斯霍尔德镜射.

(c) 证明你的算法中，在每个元素的运算包含 6 次 flop 而非 4 次，所以渐近的运算计数比（10.9）多 50%.

76

第 11 讲　最小二乘问题

自从高斯和勒让德在 1800 年左右发明了最小二乘数据拟合以来，它的应用遍及数学科学的各个分支，且已经成为十分有力的工具. 在线性代数中，这里的问题是如何求解一个矩形的即行多于列的超定方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$. 最小二乘的思想是通过最小化剩余向量 $\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}$ 的 2-范数来“解”这样的方程组.

11.1 问　　题

考虑有 n 个未知数，但 $m>n$ 的方程组，从符号方面说，希望找到一个向量 $\boldsymbol{x}\in\mathbb{C}^n$ 满足 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$，其中 $A\in\mathbb{C}^{m\times n}$，$\boldsymbol{b}\in\mathbb{C}^n$. 一般地，这样的问题没有解，只有 $\boldsymbol{b}$ 在 range($\boldsymbol{A}$) 中才会有合适的向量 $\boldsymbol{x}$ 存在，且因 $\boldsymbol{b}$ 是 m 维向量，而 range($\boldsymbol{A}$) 的维数最多是 n，这只有对 $\boldsymbol{b}$ 特殊的选择才是真实的. 我们说 $m>n$ 的矩形方程组是*超定的*（overdetermined）. 下列向量称为*剩余的*（residual）向量

$$\boldsymbol{r}=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\in\mathbb{C}^m, \tag{11.1}$$

或许可以适当的选择 x 而使其相当小，但一般不能使它等于零.

求解一个没有解的问题意味着什么呢？在超定方程组的情形，这个问题存在一个自然的答案. 因为不能使 $\boldsymbol{r}$ 为零，取而代之的是要使它尽量地小. 度量 $\boldsymbol{r}$ 的微小就必须选择一种范数. 如果选择 2-范数，问题就取为如下形式：

77

$$\begin{gathered}\text{给定 } \boldsymbol{A}\in\mathbb{C}^{m\times n}, m\geqslant n, \boldsymbol{b}\in\mathbb{C}^m,\\ \text{求 } \boldsymbol{x}\in\mathbb{C}^n\text{，使得 } \|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}\|_2 \text{ 最小化.}\end{gathered} \tag{11.2}$$

这是一般（线性）*最小二乘问题*（least squares problem）的公式表示. 2-范数的选择有各种基于几何和统计上的理由，且将看到，它最终导致简单的线性算法，因为求极小值要将二次函数的导数令为零.

2-范数对应欧几里得距离，所以（11.2）有一个简单的几何解释. 我们寻找向量 $\boldsymbol{x}\in\mathbb{C}^n$，使得向量 $\boldsymbol{A}\boldsymbol{x}\in\mathbb{C}^m$ 是在 range($\boldsymbol{A}$) 中最接近 $\boldsymbol{b}$ 的点.

11.2 例：多项式数据拟合

作为例子，我们来比较一下导出方形方程组的多项式插值与矩形方程组的最小二乘多项式数据拟合的差别.

例 11.1　多项式插值　假设给出 m 个不同的点 $x_1,\cdots,x_m\in\mathbb{C}$，以及这些点上的数据 $y_1,\cdots,y_m\in\mathbb{C}$，则存在惟一的这些点上对这些数据的*多项式插值式*（polynomial

interpolant)，它是最高次数为 $m-1$ 的多项式，

$$p(x) = c_0 + c_1 x + \cdots + c_{m-1} x^{m-1}, \tag{11.3}$$

满足性质：在每个 x_i 上，有 $p(x_i)=y_i$. 数据$\{x_i\}$,$\{y_i\}$与系数$\{c_i\}$的关系，可以用已在例 1-1 中见过的正方形范德蒙德方程组表示：

$$\begin{bmatrix} 1 & x_1 & x_1^2 & & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{m-1} \\ 1 & x_3 & x_3^2 & & x_3^{m-1} \\ & \vdots & & & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{m-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{m-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}. \tag{11.4}$$

对于给定的数据集，为了确定系数$\{c_i\}$，可以解这个方程组，只要点$\{x_i\}$是不同的就保证了方程组非奇异（习题 37.3）.

图 11-1 为一个多项式插值过程的例子. 用 × 号表示离散方波形状的 11 个数据点，曲线 $p(x)$ 必须通过它们. 然而，拟合结果根本无法令人满意，在靠近区间端点处，$p(x)$ 显示了很大的波动，这显然是插值过程人为的结果而非数据的合理反应. 78

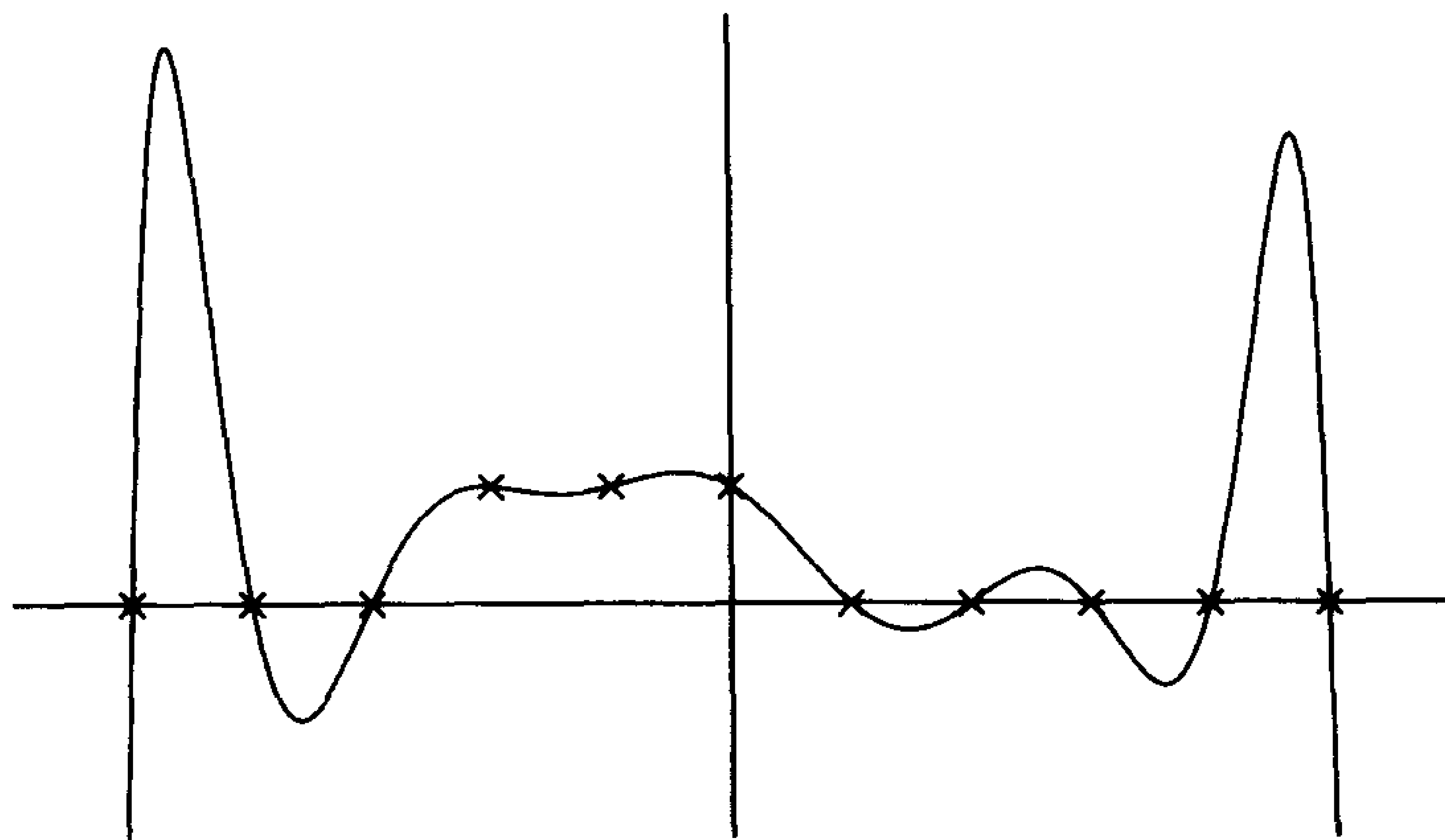


图 11-1　11 个数据点的 10 次多项式插值式. 轴的刻度没有给出，因为在图上这没有影响

这个不令人满意的行为是多项式插值的典型行为. 所作的拟合经常是不好的，如果用到更多的数据，则会导致更坏而非更好的结果. 即使拟合是好的，插值过程也可能是病态的，即对数据的扰动是灵敏的（下一讲讨论该问题）. 为避免这些问题，可以利用像在区间 [−1,1] 内切比雪夫点那样的不均匀插值点的集合. 然而，在应用上不是总能随意选择插值点的. □

例 11.2　多项式最小二乘拟合　在不改变数据点的情况下，可以通过降低多项式的次数来做得更好. 现在再次给出 $x_1,\cdots,x_m$ 和 $y_1,\cdots,y_m$，对某个 $n<m$，考虑 $n-1$ 次多项式

$$p(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1} \tag{11.5}$$

这样的多项式如果使得数据偏差的平方和

79

$$\sum_{i=1}^{m} |p(x_i) - y_i|^2. \tag{11.6}$$

最小，那么它就是对数据的最小二乘拟合. 这个平方和等于矩形范德蒙德方程组

$$\begin{bmatrix} 1 & x_1 & & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ 1 & x_3 & & x_3^{n-1} \\ & \vdots & & \vdots \\ 1 & x_m & \cdots & x_m^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}. \tag{11.7}$$

的剩余的范数平方 $\|r\|_2^2$. 图 11-2 说明，如果用 7 次多项式拟合上例中同样的 11 个数据点，将会有什么结果. 新的多项式不插值数据，但它比例 11.1 的多项式更好地抓住了数据的总体行为. 虽然在图中看不出来，但它也对扰动有较小的灵敏. □

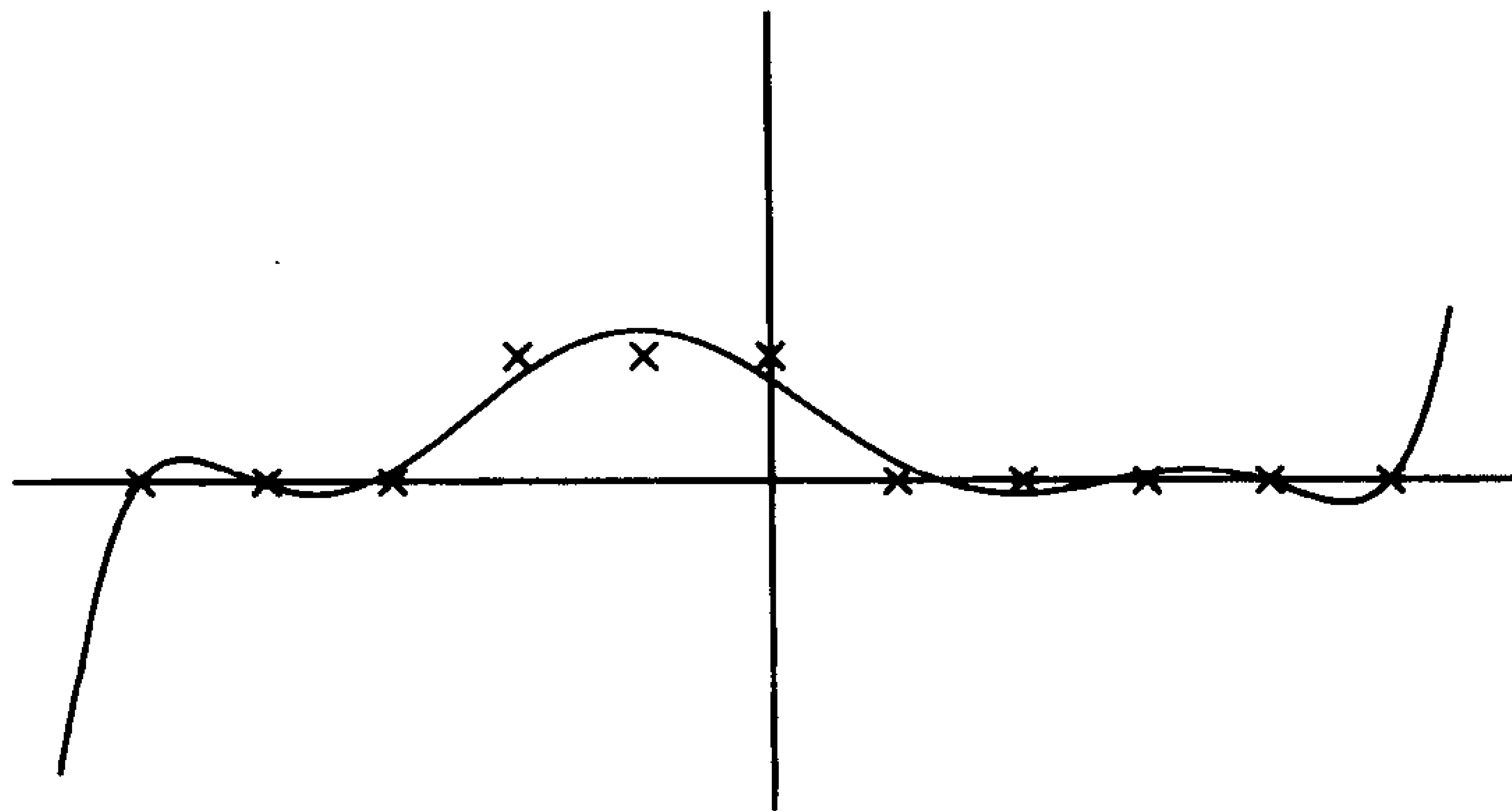


图 11-2　7 次多项式最小二乘拟合相同的 11 个数据点

11.3　正交投影和法方程组

图 11-2 的曲线是怎样计算出来的？一般如何解最小二乘问题？推导算法的关键是正交投影.

图 11-3 说明了此想法. 我们的目的是在 range(A)上找到最接近 b 的点 Ax，以便使剩余 $r = b - Ax$ 的范数最小. 从几何方面看，显然会有 $Ax = Pb$，其中 $P \in \mathbb{C}^{m\times m}$ 为映射 $\mathbb{C}^m$ 到 range(A)上的正交投影算子（第 6 讲）. 换句话说，剩余 $r = b - Ax$ 一定正交于 range(A)，这个条件写成公式就是下面的定理.

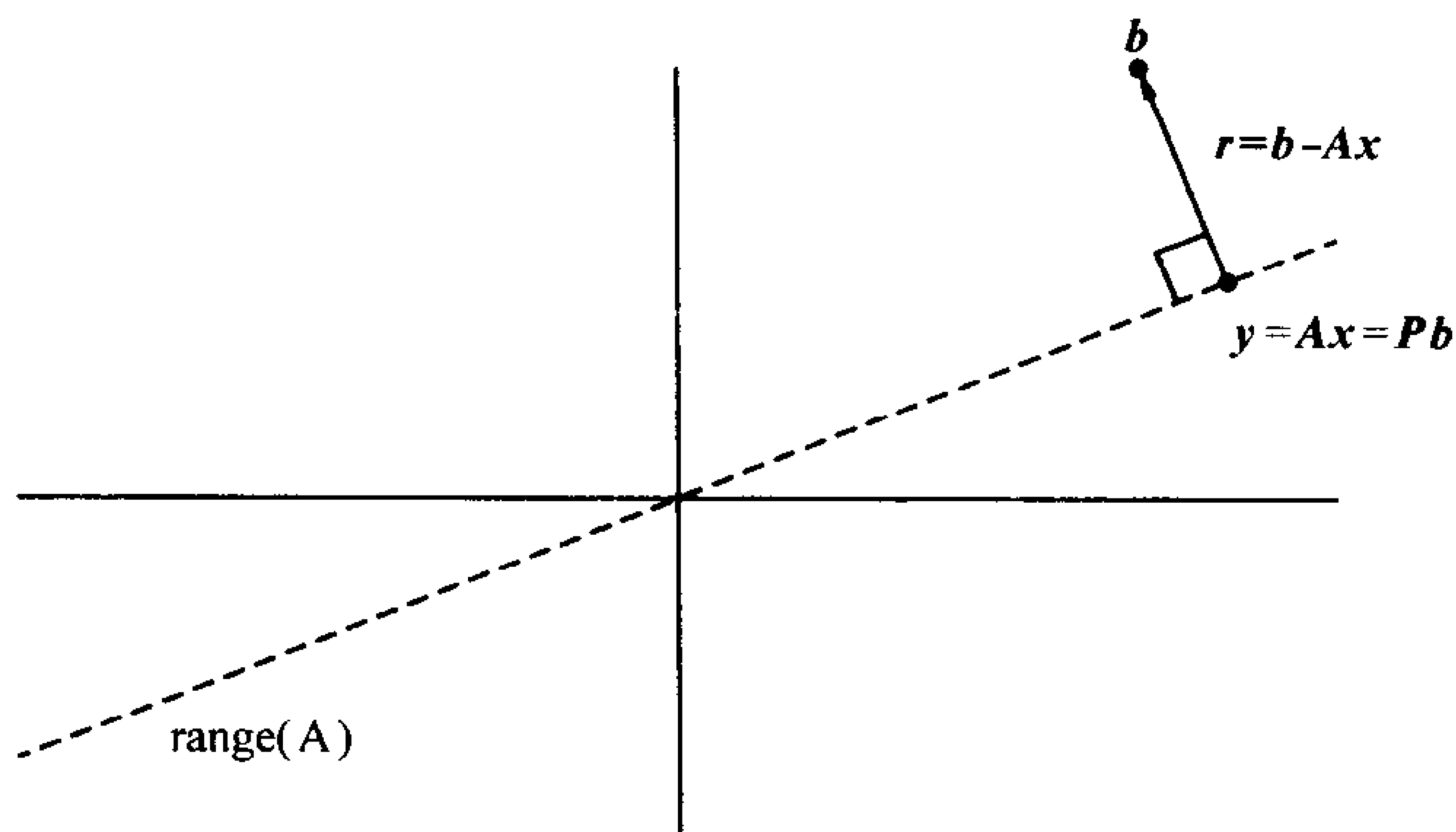


图 11-3 根据正交投影，最小二乘问题（11.2）的公式

定理 11.1 给定 $A\in\mathbb{C}^{m\times n}(m\geq n)$ 和 $b\in\mathbb{C}^{m}$，向量 $x\in\mathbb{C}^{n}$ 使剩余范数 $\|r\|_2=\|b-Ax\|_2$ 最小化，从而解出最小二乘问题（11.2），当且仅当 $r\perp\text{range}(A)$，即

$$A^*r=0,\tag{11.8}$$

80

或等价地，

$$A^*Ax=A^*b,\tag{11.9}$$

或再次等价地，

$$Pb=Ax,\tag{11.10}$$

其中 $P\in\mathbb{C}^{m\times m}$ 为到 range(A) 的正交投影算子. $n\times n$ 方程组（11.9）称为法方程组（normal equations），当且仅当 A 满秩时它是非奇异的. 进而当且仅当 A 满秩时解 x 是惟一的.

证明 （11.8）和（11.10）的等价性由在第 6 讲中讨论的正交投影算子性质可得，（11.8）和（11.9）的等价性由 r 的定义可得. 为了证明 $y=Pb$ 是在 range(A) 中最小化 $\|b-y\|_2$ 的惟一点，假设 $z\neq y$ 是在 range(A) 的另外一点，因为 $z-y$ 正交于 $b-y$，由毕达哥拉斯定理（习题 2.2）得 $\|b-z\|_2^2=\|b-y\|_2^2+\|y-z\|_2^2>\|b-y\|_2^2$. 最后，注意若 A^*A 奇异，则对某非零向量 x 有 $A^*Ax=0$，这隐含着 $x^*A^*Ax=0$（参看习题 6.3）. 因此 $Ax=0$，它隐含着 A 是秩亏损的，反之，若 A 为秩亏损，则 $Ax=0$ 对某非零的 x 成立，也隐含着 $A^*Ax=0$，因此 A^*A 奇异. 由（11.9），这个非奇异矩阵 A^*A 的特征隐含了关于 x 惟一性的陈述. □

11.4 伪 逆

刚才看到，若 A 满秩，则最小二乘问题（11.2）的解是惟一的，且由 $x=$

81

$(A^*A)^{-1}A^*b$给出. 矩阵$(A^*A)^{-1}A^*$称为A的*伪逆*（pseudoinverse），记作A^+：

$$A^+ = (A^*A)^{-1}A^* \in \mathbb{C}^{n\times m}. \tag{11.11}$$

此矩阵把向量$b \in \mathbb{C}^m$映射到向量$x \in \mathbb{C}^n$，这解释了为什么它有维数$n \times m$——列多于行.

可以综述满秩线性最小二乘问题（11.2）如下，问题是计算向量

$$x = A^+b, \qquad y = Pb, \tag{11.12}$$

中的一者或两者，其中A^+是A的伪逆，P是投影到range(A)的正交投影算子，现在描述这样做的3种主导算法.

11.5 法方程组

解最小二乘问题的经典方法是解方程组（11.9）. 若A满秩，则它是一个正方的，埃米尔特正定的n维方程组. 解这样方程组的标准方法是利用楚列斯基因子分解，这将在第23讲中讨论. 此方法构造因子分解$A^*A = R^*R$，其中R是上三角的，它将（11.9）化为方程组

$$R^*Rx = A^*b. \tag{11.13}$$

算法如下

算法 11.1　借助法方程组的最小二乘

1. 形成矩阵A^*A和向量A^*b.
2. 计算楚列斯基因子分解$A^*A = R^*R$.
3. 对w解下三角方程组$R^*w = A^*b$.
4. 对x解上三角方程组$Rx = w$.

这个计算的主要步骤是前两步（第3、第4步，参看第17讲），由于对称性，A^*A的计算只需要mn^2次flop，是若A和A^*为任意同维数矩阵情形的计算量的一半. 楚列斯基因子分解也利用对称性，需要$n^3/3$次flop. 总的来看，用法方程组解最小二乘问题包含下列全部运算计数：

82

$$\text{算法 11.1 的工作量：} \sim mn^2 + \frac{1}{3}n^3 \text{ 次 flop}. \tag{11.14}$$

11.6 QR 因子分解

自20世纪60年代开始流行的解最小二乘问题的“现代经典”方法，是基于约化QR因子分解的. 由格拉姆·施密特正交化或更常用的豪斯霍尔德三角形化可以构

造因子分解 $A=\hat{Q}\hat{R}$. 正交投影算子 P 可以写成 $P=\hat{Q}\hat{Q}^*$（见（6.6）），这样有

$$y=Pb=\hat{Q}\hat{Q}^*b. \tag{11.15}$$

因 $y\in\text{range}(A)$，所以方程组 $Ax=y$ 有一个准确解，组合 QR 因子分解和（11.15）给出

$$\hat{Q}\hat{R}x=\hat{Q}\hat{Q}^*b, \tag{11.16}$$

左乘 $\hat{Q}^*$，得到结果

$$\hat{R}x=\hat{Q}^*b. \tag{11.17}$$

（现在乘以 $\hat{R}^{-1}$ 就得出伪逆公式 $A^+=\hat{R}^{-1}\hat{Q}^*$.）方程（11.17）是一个上三角方程组，若 A 满秩，则它是非奇异的，容易由向后代入求解（第 17 讲）.

算法 11.2　借助 QR 因子分解的最小二乘

1. 计算约化 QR 因子分解 $A=\hat{Q}\hat{R}$.
2. 计算向量 $\hat{Q}^*b$.
3. 对 x 解上三角方程组 $\hat{R}x=\hat{Q}^*b$.

注意，（11.17）也可由法方程组推导. 若 $A^*Ax=A^*b$，则 $\hat{R}^*\hat{Q}^*\hat{Q}\hat{R}x=\hat{R}^*\hat{Q}^*b$，这隐含了 $\hat{R}x=\hat{Q}^*b$.

算法 11.2 的工作量主要由 QR 因子分解的代价决定，如果豪斯霍尔德镜射算子用于此步，由（10.9），有

$$\text{算法 11.2 的工作量：}\sim 2mn^2-\frac{2}{3}n^3 \text{ 次 flop.} \tag{11.18}$$

11.7　SVD

第 31 讲中我们将描述计算约化奇异值分解 $A=\hat{U}\hat{\Sigma}V^*$ 的算法. 这启示了解最小二乘问题的另一种方法. 现在 P 表示为 $P=\hat{U}\hat{U}^*$，给出

$$y=Pb=\hat{U}\hat{U}^*b, \tag{11.19}$$

与（11.16）和（11.17）类似的结果是

$$\hat{U}\hat{\Sigma}V^*x=\hat{U}\hat{U}^*b \tag{11.20}$$

及

$$\hat{\Sigma} V^* x = \hat{U}^* b. \tag{11.21}$$

(乘以 $V\hat{\Sigma}^{-1}$就得出 $A^+ = V\hat{\Sigma}^{-1}\hat{U}^*$.) 算法看起来是这样：

算法 11.3　借助 SVD 的最小二乘

1. 计算约化 SVD $A = \hat{U}\hat{\Sigma}V^*$.
2. 计算向量$\hat{U}^* b$.
3. 对 w 解对角方程组 $\hat{\Sigma} w = \hat{U}^* b$.
4. 令 $x = Vw$.

注意，QR 因子分解把最小二乘问题化为一个三角方程组，而 SVD 把它化为一个对角方程组，当然很容易解它. 若 A 满秩，则对角方程组是非奇异的.

如前，(11.21) 可由法方程组导出. 若 $A^*Ax = A^*b$，则 $V\hat{\Sigma}^*\hat{U}^*\hat{U}\hat{\Sigma}V^*x = V\hat{\Sigma}^*\hat{U}^*b$，隐含着 $\hat{\Sigma}V^*x = \hat{U}^*b$.

算法 11.3 的运算计数主要由 SVD 的计算量决定. 在第 31 讲中将看到，对 $m \gg n$，其代价近似地与 QR 因子分解相同，但对 $m \approx n$，SVD 的计算量更多些. 典型的估计是

$$\text{算法 11.3 的工作量：} \sim 2mn^2 + 11n^3 \text{ 次 flop}, \tag{11.22}$$

对此结果的判定参看第 31 讲.

11.8　算法的比较

我们描述过的每种方法在某些情况下各有其优越性. 当速度是惟一的考虑时，算法 11.1 可能是最好的. 然而，由于舍入误差的出现，解法方程组不总是稳定的，因此多年来，数值分析学家推荐用算法 11.2 替代它作为解最小二乘问题的标准方法. 诚然它是一个自然优美的算法，我们推荐它作为“日常应用”的方法. 然而，如果 A 接近秩亏损时，算法 11.2 稳定性质不理想，在这种情形下，有很好的理由转而使用基于 SVD 的算法 11.3.

使一个算法在某些情况而非其他情况下好于另一个算法的稳定性考虑因素是什么呢？现在是系统地讨论这些事情的时候了. 我们将在第 18 讲和第 19 讲中转回到最小二乘问题算法的研究.

84

习　题

11.1　设 $m \times n$ 矩阵 A 有形式

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix},$$

其中，A_1 是维数 $n \times n$ 的非奇异矩阵，A_2 是维数 $(m-n) \times n$ 的任意矩阵. 证明 $\|A^+\|_2 \leqslant \|A_1^{-1}\|_2$.

11.2 (a) 作为在区间 [1,2] 上的 L^2 范数度量，用函数 e^x，$\sin x$ 和 $\Gamma(x)$ 的线性组合，怎样紧密地拟合函数 $f(x) = x^{-1}$？（$\Gamma(x)$ 是伽马函数，MATLAB 中的一种内置函数.）用 [1,2] 的离散化和离散最小二乘问题写出求至少有两位相对准确数字答案的程序. 写出你的答案估计和最优线性组合的系数，并做出最优逼近图.

(b) 将 [1,2] 改为 [0,1]，重复以上内容. 你可以发现以下有用的事实：若 $g(x) = 1/\Gamma(x)$，则 $g'(0) = 1$.

11.3 取 $m = 50, n = 12$，用 MATLAB 的 `linspace` 定义 t 为对应于由 0 到 1 的直线间隔格子点的 m 维向量. 用 MATLAB 的 `vander` 和 `fliplr` 定义 A 为，与在这些格子点上以 $n-1$ 次多项式最小二乘拟合相关的 $m \times n$ 矩阵. 取 b 为函数 $\cos(4t)$ 在格子点上的值. 现在用 6 种方法计算并输出（用十六位精度）最小二乘系数向量 x：

(a) 法方程组的公式和解，用 MATLAB 的 `\`，

(b) 用 `mgs`（修正格拉姆-施密特，习题 8.2）作 QR 因子分解计算，

(c) 用 `house`（豪斯霍尔德三角形化，习题 10.2）作 QR 因子分解计算，

(d) 用 MATLAB 的 `qr`（也是豪斯霍尔德三角形化）作 QR 因子分解计算，

(e) MATLAB 中的 `x = A\b`（也基于 QR 因子分解），

(f) SVD，用 MATLAB 中的 `svd`.

(g) 上述计算将产生 12 个系数的 6 个表格，在每个表格上，用红笔标出出现错误（受舍入误差影响）的数字. 说明你观察到的差异. 法方程组是否显示出不稳定性？不必解释你的观测.

第Ⅲ部分

条件和稳定性

第 12 讲　条件和条件数

在本书的第Ⅲ部分，我们将系统讨论数值分析的两个基本问题，这是直到现在我们还回避的问题. 条件（conditioning）与数学问题的扰动行为有关. 稳定性（stability）与在计算机上解这些问题所用算法的扰动行为有关.

12.1　问题的条件

抽象地说，可以将一个问题看成一个由数据的赋范向量空间 $\boldsymbol{X}$ 到解的赋范向量空间 $\boldsymbol{Y}$ 的函数 $f:X\to Y$. 这个函数 f 通常是非线性的（甚至在线性代数中），但在大多数情况下至少是连续的.

我们通常关注问题 f 在一个特殊的数据点 $x\in X$ 上的行为（不同点的行为可能有非常大的变化）. 一个问题 f 与描述的数据 $\boldsymbol{x}$ 的组合可以称为一个问题例子，但是这个概念更通用的术语仍是问题，虽然有时会引起混乱.

良态（well-conditioned）问题（例子）是具有 $\boldsymbol{x}$ 的所有小扰动只导致 $f(\boldsymbol{x})$ 小改变这样性质的问题. 病态（ill-conditioned）问题是具有 $\boldsymbol{x}$ 的某些小扰动导致 $f(\boldsymbol{x})$ 大改变这样性质的问题.

在这些叙述中，“大”和“小”的意义依赖于实际应用. 特别地，有时最适合按一个绝对尺度度量扰动，而有时则最适合相对于被扰动的对象的范数去度量它们.

12.2　绝对条件数

令 $\delta\boldsymbol{x}$ 记为 $\boldsymbol{x}$ 的一个小扰动，写成 $\delta f=f(\boldsymbol{x}+\delta\boldsymbol{x})-f(\boldsymbol{x})$. 问题 f 在 $\boldsymbol{x}$ 的绝对条件数（absolute condition number）$\hat{\kappa}=\hat{\kappa}(\boldsymbol{x})$ 定义为

$$\hat{\kappa}=\lim_{\delta\to 0}\sup_{\|\delta x\|\leqslant\delta}\frac{\|\delta f\|}{\|\delta\boldsymbol{x}\|}. \tag{12.1}$$

对大多数问题，此公式中上确界的极限可以解释为遍及所有无穷小扰动 $\delta\boldsymbol{x}$ 的上确界，为了可读方面的考虑，一般将公式简写为

$$\hat{\kappa}=\sup_{\delta x}\frac{\|\delta f\|}{\|\delta\boldsymbol{x}\|}, \tag{12.2}$$

其中理解 $\delta\boldsymbol{x}$ 和 δf 为无穷小.

若 f 可微，可以用 f 的导数计算条件数. 令 $\boldsymbol{J}(\boldsymbol{x})$ 是其 i,j 元素为偏导数 $\partial f_i/\partial x_j$ 在 $\boldsymbol{x}$ 点的值的矩阵，即 f 在 $\boldsymbol{x}$ 的雅可比矩阵. 导数的定义给出一阶近似 $\delta f\approx\boldsymbol{J}(\boldsymbol{x})\delta\boldsymbol{x}$，且极

限$\|\delta x\|\to 0$时取等号．绝对条件数成为

$$\hat{\kappa}=\|J(x)\|, \tag{12.3}$$

其中$\|J(x)\|$表示由在X和Y上的范数导出的$J(x)$的范数．

12.3　相对条件数

当我们关心相对变化时，需要相对条件的概念．相对条件数（relative condition number）$\kappa=\kappa(x)$定义为

$$\kappa=\lim_{\delta\to 0}\sup_{\|\delta x\|\leqslant\delta}\left(\frac{\|\delta f\|}{\|f(x)\|}\Big/\frac{\|\delta x\|}{\|x\|}\right), \tag{12.4}$$

或者再次假设δx和δf是无穷小，写成

$$\kappa=\sup_{\delta x}\left(\frac{\|\delta f\|}{\|f(x)\|}\Big/\frac{\|\delta x\|}{\|x\|}\right). \tag{12.5}$$

若f可微，可按照雅可比矩阵将此量表示为

90

$$\kappa=\frac{\|J(x)\|}{\|f(x)\|/\|x\|}. \tag{12.6}$$

绝对和相对条件数两者都有它们的应用，但后者在数值分析中更为重要．因为归根到底用在计算机的浮点运算引入的是相对误差而非绝对误差，参看下一讲．如果κ很小（例如，1，10，10^2），则问题是良态的，如果κ很大（例如，10^6，10^{16}），则问题是病态的．

12.4　例　　子

例 12.1　考虑由$x\in\mathbb{C}$得到数量$x/2$的平凡例子．函数f：$x\mapsto x/2$的雅可比矩阵正好是导数$J=f'=1/2$，所以由（12.6），

$$\kappa=\frac{\|J\|}{\|f(x)\|/\|x\|}=\frac{1/2}{(x/2)/x}=1.$$

以任何标准来看，这个问题都是良态的．□

例 12.2　考虑对$x>0$计算$\sqrt{x}$的问题．f：$x\mapsto\sqrt{x}$的雅可比矩阵是导数$J=f'=1/(2\sqrt{x})$，所以有

$$\kappa=\frac{\|J\|}{\|f(x)\|/\|x\|}=\frac{1/(2\sqrt{x})}{\sqrt{x}/x}=\frac{1}{2}.$$

它是又一个良态问题. □

例 12.3　考虑由向量 $\boldsymbol{x}=(x_1,x_2)^*\in\mathbb{C}^2$ 得到数量 $f(\boldsymbol{x})=x_1-x_2$ 的问题. 为了简单起见，在数据空间$\mathbb{C}^2$ 中使用∞-范数. f 的雅可比矩阵是

$$\boldsymbol{J}=\left[\frac{\partial f}{\partial x_1}\quad\frac{\partial f}{\partial x_2}\right]=[1\quad -1],$$

且$\|\boldsymbol{J}\|_\infty=2$. 因而条件数为

$$\kappa=\frac{\|\boldsymbol{J}\|_\infty}{\|f(\boldsymbol{x})\|/\|\boldsymbol{x}\|}=\frac{2}{|x_1-x_2|/\max\{|x_1|,|x_2|\}}.$$

如果$|x_1-x_2|\approx 0$，则 κ 是大的，所以当 $x_1\approx x_2$ 时，问题是病态的，这与我们对“消去误差”危害的直觉是符合的. □

例 12.4　考虑 x 接近 10^{100}时 $f(x)=\tan x$ 的计算. 在这个问题中，x 微小的相对扰动可引起 $\tan x$ 任意大的改变. 结论是：$\tan(10^{100})$在大多数计算机上不能有效地计算. 同样微小的扰动引起 $\tan x$ 的导数的任意改变，所以在试图计算雅可比矩阵而不是观察它不是小量的时候，有些细节可以考虑. 有一个故事，其中的精彩之处精确地依赖于 $\tan(10^{100})$ 的病态，参看 Richard Feynman 的书 *Surely You're Joking, Mr. Feynman*，中文名为《别闹了，费曼先生》中的“幸运数”一章. □　91

例 12.5　给定一个多项式的系数去求它的根，这是病态问题的一个经典例子. 考虑 $x^2-2x+1=(x-1)^2$，在 $x=1$ 有重根. 系数小的扰动会导致根大的改变；例如 $x^2-2x+0.9999=(x-0.99)(x-1.01)$. 事实上，根可以按系数变化的平方根的比例改变，在此情形下，雅可比矩阵是无限的（此问题不可微），$\kappa=\infty$.

甚至在不含重根的情形，多项式求根也是典型病态的. 如果多项式 $p(x)$ 的第 i 个系数 a_i 被无穷小量 δa_i 扰动，第 j 个根 x_j 的扰动是 $\delta x_j=-(\delta a_i)x_j^i/p'(x_j)$，其中 p' 记为 p 的导数. 对于单个系数 a_i 的扰动，x_j 的条件数是

$$\kappa=\frac{|\delta x_j|}{|x_j|}\Bigg/\frac{|\delta a_i|}{|a_i|}=\frac{|a_ix_j^{i-1}|}{|p'(x_j)|}. \tag{12.7}$$

这个数经常是很大的. 考虑“Wilkinson 多项式”：

$$p(x)=\prod_{i=1}^{20}(x-i)=a_0+a_1x+\cdots+a_{19}x^{19}+x^{20}. \tag{12.8}$$

此多项式最灵敏的根是 $x=15$，它对系数 $a_{15}\approx 1.67\times10^9$ 中的改变最灵敏. 条件数为

$$\kappa\approx\frac{1.67\times10^9\cdot 15^{14}}{5!\ 14!}\approx 5.1\times10^{13}.$$

图 12-1 用图形说明了病态. □

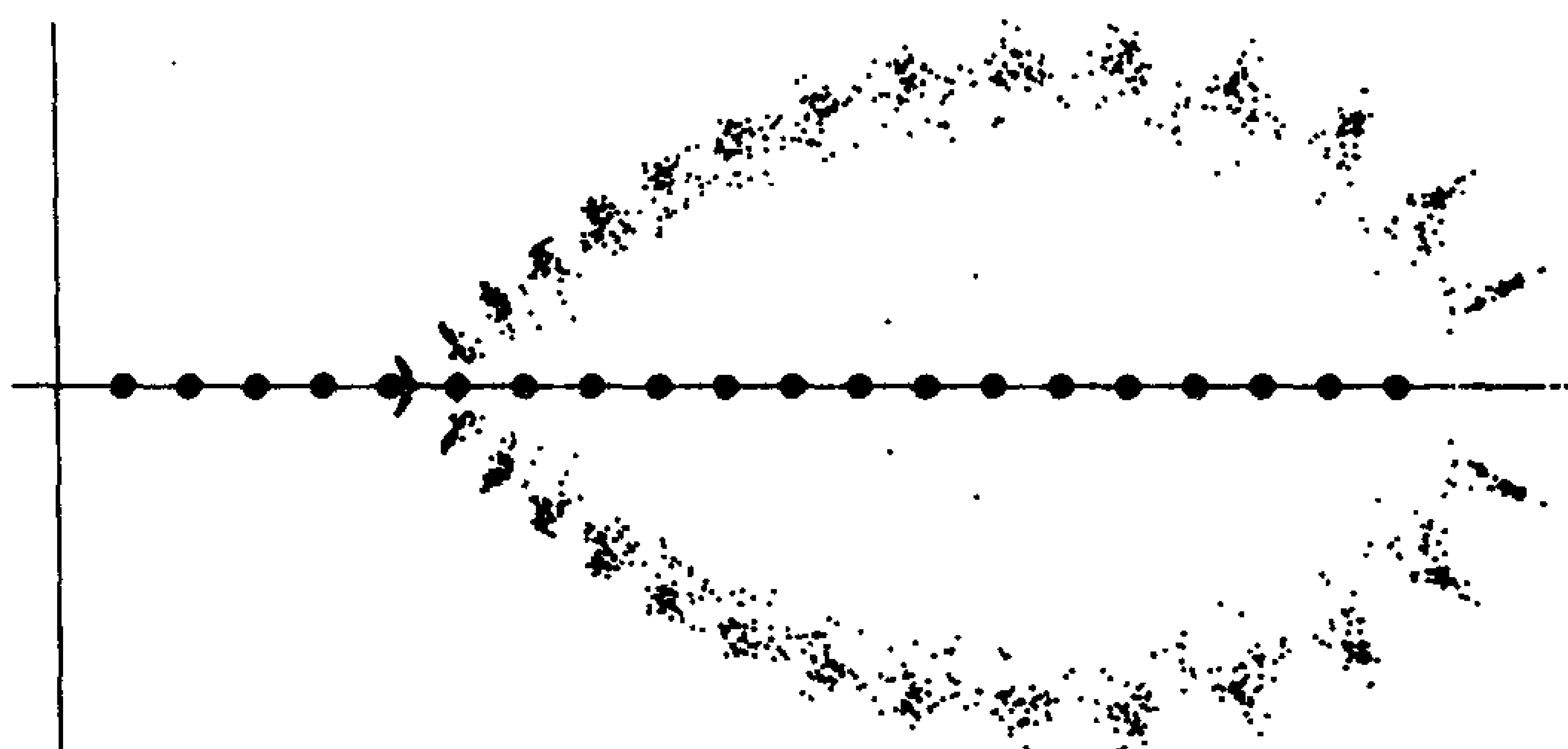


图 12-1　Wilkinson 的经典病态例子. 大的点是没有扰动的多项式（12.8）的根，小的点是在复平面上系数定义为 $\tilde{a}_k = a_k(1+10^{-10}r_k)$ 的 100 个随机扰动多项式的重叠的根，其中 r_k 是由均值为 0，方差为 1 的正态分布得来的数

例 12.6　计算非对称矩阵特征值的问题也通常是病态的. 可以再比较两个矩阵

$$\begin{bmatrix} 1 & 1000 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1000 \\ 0.001 & 1 \end{bmatrix},$$

中看到这一点，这两个矩阵的特征值分别是{1，1}和{0，2}. 另一方面，如果矩阵 A 是对称的（更一般地，如果它是正规的），则它的特征值是良态的. 可以证明若 λ 和 $\lambda+\delta\lambda$ 是 A 和 $A+\delta A$ 对应的特征值，则 $|\delta\lambda| \leq \|\delta A\|_2$，其中当 δA 是单位矩阵的倍数时等式成立（习题 26.3）. 因此如果扰动用 2-范数度量，对称特征值问题的绝对条件数为 $\hat{\kappa}=1$，相对条件数为 $\kappa = \|A\|_2/|\lambda|$. □

92

12.5　矩阵-向量乘法的条件

现在转到数值线性代数中最重要的条件数之一的讨论.

固定 $A \in \mathbb{C}^{m\times n}$，考虑由输入 x 计算 Ax 的问题，也即我们要确定对应于 x 的扰动而 A 没有扰动的条件数. 以 $\|\cdot\|$ 记为任意的向量范数及对应的导出矩阵范数，由 κ 的定义直接得

$$\kappa = \sup_{\delta x}\left(\frac{\|A(x+\delta x)-Ax\|}{\|Ax\|} \Big/ \frac{\|\delta x\|}{\|x\|} \right) = \sup_{\delta x} \frac{\|A\delta x\|}{\|\delta x\|} \Big/ \frac{\|Ax\|}{\|x\|},$$

即是

$$\kappa = \|A\|\frac{\|x\|}{\|Ax\|} \tag{12.9}$$

[（12.6）的一个特殊情形]. 这是 κ 的一个准确公式，它依赖于 A 和 x 两者.

假设以上的计算中 A 是正方形矩阵且非奇异，因此可用 $\|x\|/\|Ax\| \leq \|A^{-1}\|$ 这一

事实把（12.9）放大到一个不依赖于 $\boldsymbol{x}$ 的界：

$$\kappa \leqslant \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|. \tag{12.10}$$

或可写成

$$\kappa = \alpha\|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\| \tag{12.11}$$

93

且

$$\alpha = \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|}\Big/\|\boldsymbol{A}^{-1}\|. \tag{12.12}$$

对于 $\boldsymbol{x}$ 的某些选择，有 $\alpha = 1$，因而 $\kappa = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$. 如果 $\|\cdot\| = \|\cdot\|_2$，则这会出现在 $\boldsymbol{x}$ 是 $\boldsymbol{A}$ 的最小右奇异向量的倍数的情形.

事实上，$\boldsymbol{A}$ 不必是正方的. 若 $\boldsymbol{A} \in \mathbb{C}^{m\times n}(m \geqslant n)$满秩，用定义于（11.11）的伪逆 $\boldsymbol{A}^+$替代 $\boldsymbol{A}^{-1}$时，方程（12.10）~(12.12）仍然成立.

反问题会怎么样：给定 $\boldsymbol{A}$，由输入 $\boldsymbol{b}$ 计算 $\boldsymbol{A}^{-1}\boldsymbol{b}$. 从数学上看，这等同于刚才的问题，只是 $\boldsymbol{A}$ 代之以 $\boldsymbol{A}^{-1}$. 因此我们已经证明了下面的定理.

定理 12.1　令 $\boldsymbol{A} \in \mathbb{C}^{m\times m}$非奇异，考虑方程 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. 给定 $\boldsymbol{x}$，计算 $\boldsymbol{b}$ 的问题对于 $\boldsymbol{x}$ 的扰动有条件数

$$\kappa = \|\boldsymbol{A}\|\frac{\|\boldsymbol{x}\|}{\|\boldsymbol{b}\|} \leqslant \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|. \tag{12.13}$$

给定 $\boldsymbol{b}$，计算 $\boldsymbol{x}$ 的问题对于 $\boldsymbol{b}$ 的扰动有条件数

$$\kappa = \|\boldsymbol{A}^{-1}\|\frac{\|\boldsymbol{b}\|}{\|\boldsymbol{x}\|} \leqslant \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|. \tag{12.14}$$

若$\|\cdot\| = \|\cdot\|_2$，则如果 $\boldsymbol{x}$ 是 $\boldsymbol{A}$ 对应于最小奇异值 σ_m 的右奇异向量的倍数时，(12.13）中等号成立，且如果 $\boldsymbol{b}$ 是 $\boldsymbol{A}$ 对应最大奇异值 σ_1 的左奇异向量的倍数时，(12.14）中等号成立.

12.6　矩阵的条件数

乘积$\|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$如此频繁地出现使它有了自己的名字：它是 $\boldsymbol{A}$（对于范数$\|\cdot\|$）的条件数，记为 $\kappa(\boldsymbol{A})$：

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|. \tag{12.15}$$

因此在这种情形下，术语“条件数”附属于一个矩阵而非一个问题. 如果 $\kappa(\boldsymbol{A})$小，$\boldsymbol{A}$ 称为良态的；如果 $\kappa(\boldsymbol{A})$大，$\boldsymbol{A}$ 是病态的. 若 $\boldsymbol{A}$ 奇异，习惯写成 $\kappa(\boldsymbol{A}) = \infty$.

注意，若$\|\cdot\| = \|\cdot\|_2$，则$\|\boldsymbol{A}\| = \sigma_1$，$\|\boldsymbol{A}^{-1}\| = 1/\sigma_m$. 因此 2-范数意义如下

$$\kappa(A)=\frac{\sigma_1}{\sigma_m},\tag{12.16}$$

94 这个公式一般用于计算矩阵的2-范数条件数. 比值σ_1/σ_m可以解释为，$\mathbb{C}^m$中单位球面在A作用下的象的那个超椭圆（图4-1）的离心率.

对于满秩的矩形矩阵$A\in\mathbb{C}^{m\times n}$（$m\geqslant n$），条件数根据伪逆来定义：$\kappa(A)=\|A\|\|A^+\|$. 因为$A^+$是由最小二乘问题的启示而得，这个定义在$\|\cdot\|=\|\cdot\|_2$的情形最有用，在此情形有

$$\kappa(A)=\frac{\sigma_1}{\sigma_n}.\tag{12.17}$$

12.7　方程组的条件

在定理12.1中，我们令A固定而扰动x或b. 如果我们扰动A则会怎样？特别地，令b固定而考虑当A有无穷小扰动δA时问题$A\mapsto x=A^{-1}b$的行为. 这样x一定有无穷小的改变δx，其中

$$(A+\delta A)(x+\delta x)=b.$$

利用等式$Ax=b$并舍弃双重无穷小项$(\delta A)(\delta x)$，得到$(\delta A)x+A(\delta x)=\mathbf{0}$，即$\delta x=-A^{-1}(\delta A)x$. 这个方程隐含$\|\delta x\|\leqslant\|A^{-1}\|\|\delta A\|\|x\|$，或等价地

$$\frac{\|\delta x\|}{\|x\|}\bigg/\frac{\|\delta A\|}{\|A\|}\leqslant\|A^{-1}\|\|A\|=\kappa(A).$$

当δA使得

$$\|A^{-1}(\delta A)x\|=\|A^{-1}\|\|\delta A\|\|x\|,$$

成立时，上述式子在其界上等号成立，且可用对偶范数（习题3.6）证明，对任意的A和b及范数$\|\cdot\|$，存在这样的扰动δA. 这导出下面的结果.

定理12.2　令b固定并考虑计算$x=A^{-1}b$的问题，其中A是正方且非奇异的. 这个问题对于A扰动的条件数是

$$\kappa=\|A\|\|A^{-1}\|=\kappa(A).\tag{12.18}$$

定理12.1和定理12.2在数值线性代数中十分重要，因为它决定了人们能够怎样准确地解方程组，如果问题$Ax=b$包含了一个病态矩阵A，则除了在非常特殊的情况下，在计算解时总一定会预料到“丧失$\log_{10}\kappa(A)$个数字位”. 稍后我们将重新讨论这种现象，对最小二乘问题类似的结果将在第18讲中讨论. 95

习 题

12.1 假设 A 是一个 202×202 矩阵，且 $\|A\|_2=100$，$\|A\|_F=101$. 给出 2-范数条件数 $\kappa(A)$ 可能的最清晰的下界.

12.2 在例 11.1 中观察到等距点多项式插值是病态的. 为了说明这个现象，令 $x_1,\cdots,x_n$ 和 $y_1,\cdots,y_m$ 分别为由 -1 到 1 的 n 个和 m 个等距点.

(a) 导出将数据 $\{x_j\}$ 的 n 维向量映射到样本值 $\{p(y_j)\}$ 的 m 维向量的 $m\times n$ 矩阵 A.

(b) 写出计算 A 的程序，并对 $n=1,2,\cdots,30, m=2n-1$ 作出 $\|A\|_\infty$ 半对数尺度的图. 在连续极限 $m\to\infty$ 的情形，数 $\|A\|_\infty$ 称为等距插值的勒贝格常数，当 $n\to\infty$ 时，它渐近于 $2^n/(e(n-1)\log n)$.

(c) 对 $n=1,2,\cdots,30$ 和 $m=2n-1$，插值常数函数 1 的问题的 ∞-范数条件数是什么? 利用 (12.6).

(d) 对 $n=11$，你的结果与隐含在图 11-1 的界是如何接近的?

12.3 本习题的目的是考察随机矩阵的某些性质. 你的任务是作为一个实验科学家，完成一些能导出猜想和更精细实验的实验. 不要企图证明任何事情，作出精心设计的有 1000 个数的图形.

定义随机矩阵 (random matrix) 为一个 $m\times m$ 矩阵，其元素是由均值为零，标准差为 $m^{-1/2}$ 的实正态分布产生的独立样本. (在 MATLAB 中，`A = randn(m,m)/sqrt(m)`.) 引入因子 $\sqrt{m}$，使得当 $m\to\infty$ 时其极限行为清晰可见.

(a) 随机矩阵的特征值看起来像什么? 如果你取 100 个随机矩阵并把它们的特征值重叠在单个的图中，将会怎样? 如果你对 $m=8,16,32,64,\cdots$ 这样做，显示出什么样的图案? 当 $m\to\infty$ 时谱半径 $\rho(A)$ (习题 3.2) 如何表现?

(b) 范数会怎样? 当 $m\to\infty$ 时随机矩阵的 2-范数如何表现? 当然，一定会有 $\rho(A)\leqslant\|A\|$ (习题 3.2)，当 $m\to\infty$ 时这个不等式是否有接近等式的趋势?

(c) 条件数 (或更简单地，最小奇异值 $\sigma_{\min}$) 会怎样? 甚至对固定的 m，这个问题也是有趣的. $\mathbb{R}^{m\times m}$ 中随机矩阵有 $\sigma_{\min}\leqslant 2^{-1},4^{-1},8^{-1},\cdots$ 的比例大概是多少? 换句话说，最小奇异值概率分布的尾部像什么? 所有这些量随 m 如何改变?

(d) 如果考虑用随机三角形矩阵代替满矩阵，(a)~(c) 的答案如何变化? 即考虑其元素为由上述同样分布产生的样本所组成上三角形矩阵的情形.

96

第13讲　浮点运算

计算机发明之后不久，对于如何在数字机上用恰当的方法表示实数形成了一致的意见. 诀窍是浮点运算，它是科学记数法的硬件模拟，在开始研究数值线性代数算法的准确性之前，我们一定要仔细考察这个课题.

13.1　数字表示法的限制

因为数字计算机用有限个二进制位数表示一个实数，它们只能表示实数（或复数，在本讲末尾讨论）的一个有限子集. 这个限制带来了两个困难. 第一，所表示的数不能任意大或任意小. 第二，它们之间一定存在间隙.

现代计算机表示的数已经足够大或足够小了，因此第一个制约很少造成困难. 例如，广泛应用的IEEE双精度运算允许数大到1.79×10^{308}和小到2.23×10^{-308}，这对本书所考虑的大多数问题是一个足够大的区域. 换句话说，上溢（overflow）和下溢（underflow）通常不是严重的危害（但是，如果想要计算行列式的值的话就要密切注意了!）.

与之对比，所表示的数之间存在间隙的问题是一个影响整个科学计算的问题. 例如，在IEEE双精度运算中，区间［1,2］表示为离散子集

$$1,1+2^{-52},1+2\times2^{-52},1+3\times2^{-52},\cdots,2. \tag{13.1}$$

区间［2,4］表示为同样的数乘以2，

$$2,2+2^{-51},2+2\times2^{-51},2+3\times2^{-51},\cdots,4,$$

一般地，区间［$2^j,2^{j+1}$］表示为（13.1）乘2^j. 因此在IEEE双精度运算中，相邻数之间的间隙在相对意义下永不大于$2^{-52}\approx2.22\times10^{-16}$. 这似乎可以忽略，如果使用稳定的算法（参看下一讲），对大多数实用的场合是这样的. 但是使人惊奇的是，有太多粗心大意构造的算法是不稳定的!

13.2　浮　点　数

IEEE运算是基于实数的一个浮点表示法的运算系统的一个例子. 这是当今通用计算机的通例. 在浮点数系统中，十进制（二进制）点的位置由数字分开存储，相邻两个所表示的数之间的间隙与数的大小成比例. 这不同于定点（fixed point）表示法，那里所有的间隙都有相同的大小.

特别地，考虑如下定义的一个理想化的浮点数系统，这系统由一个实数$\mathbb{R}$的离散

子集 $\mathbf{F}$ 组成，它由整数 $\beta \geqslant 2$ 和 $t \geqslant 1$ 所决定，β 称为基数（base 或 radix）（一般为 2），t 称为精度（precision）（对 IEEE 单精度和双精度分别为 24 和 53）. $\mathbf{F}$ 的元素是数 0 和形如

$$x = \pm (m/\beta^t)\beta^e \tag{13.2}$$

的所有数的全体，其中 m 是区域 $1 \leqslant m \leqslant \beta^t$ 中的一个整数，e 是任意的整数. 等价地，可以限制区域为 $\beta^{t-1} \leqslant m \leqslant \beta^t - 1$，而使 m 的选择是惟一的. 量 $\pm(m/\beta^t)$ 称作 x 的尾数（fraction 或 mantissa），而 e 则是指数.

我们的浮点系统是理想化的，它忽略了上溢和下溢. 因此，$\mathbf{F}$ 是一个可数的无限集，且它是自相似的：$\mathbf{F} = \beta\mathbf{F}$.

13.3 机器 ϵ

$\mathbf{F}$ 的分辨率习惯上由称为机器ϵ的数来概括. 暂时定义这个数为

$$\epsilon_{机器} = \frac{1}{2}\beta^{1-t}. \tag{13.3}$$

（在（13.7）之后将修正这个定义.）这个数是 1 和下一个大于它的浮点数之间的距离之半. 在相对意义下，这是和浮点数之间的间隙同样大的，也就是$\epsilon_{机器}$有下列性质： 98

$$对所有\ x \in \mathbb{R}, 存在\ x' \in \mathbf{F}, 使得\ |x - x'| \leqslant \epsilon_{机器}|x|. \tag{13.4}$$

对于共同在各种计算机上的 β 和 t 的值，通常$\epsilon_{机器}$位于 10^{-6} 和 10^{-35} 之间，在 IEEE 单精度和双精度运算中，$\epsilon_{机器}$分别规定为 $2^{-24} \approx 5.96 \times 10^{-8}$ 和 $2^{-53} \approx 1.11 \times 10^{-16}$.

令 fl：$\mathbb{R} \to \mathbf{F}$ 为对一个实数给出的最接近的浮点近似的函数，即实数在浮点系统的舍入（rounded）等价.（为了我们的目的，连接可以任意中断，虽然为了避免统计偏差，但连接的处理本身是有意义的事情.）不等式（13.4）可以用 fl 来建立：

$$\begin{gathered}对一切\ x \in \mathbb{R}, 存在满足\ |\epsilon| \leqslant \epsilon_{机器}\ 的\ \epsilon, \\ 使得\ \mathrm{fl}(x) = x(1+\epsilon).\end{gathered} \tag{13.5}$$

也就是一个实数和最接近它的浮点近似之间的差在相对意义上总小于$\epsilon_{机器}$.

13.4 浮点运算

当然，光是表示实数还不够，还要用它们来计算. 在一台计算机上，所有的数学计算都化为一些初等的算术运算，其中经典的是 +，−，× 和 ÷. 从数学上看，这些符号表示在$\mathbb{R}$上的运算. 在一台计算机上，它们有其模拟，即在 $\mathbf{F}$ 上的运算. 习惯上记这些浮点运算为$\oplus$，$\ominus$，$\otimes$和$\oslash$.

一台计算机一定要建立在下列设计原则的基础上．令 x 和 y 为任意的浮点数，即 $x, y \in \mathbf{F}$，令 $*$ 为运算 $+$，$-$，$\times$ 或 $\div$ 之一，且令 $\circledast$ 为其浮点模拟．则 $x \circledast y$ 一定要由

$$x \circledast y = \mathrm{fl}(x * y). \tag{13.6}$$

准确地给出．如果这个性质成立，则由（13.5）和（13.6）可归结出计算机有一个简单而有力的性质．

浮点运算的基本公理

对所有 $x, y \in \mathbf{F}$，存在满足 $|\epsilon| \leq \epsilon_{机器}$ 的 ϵ，使得

$$x \circledast y = (x * y)(1 + \epsilon). \tag{13.7}$$

99 从字面上看，*浮点算术运算的每个运算相对误差为* $\epsilon_{机器}$.

13.5　机器 ϵ 再讨论

本书舍入误差分析是基于（13.5）和（13.7）的，而不是基于上述浮点运算的其他细节．这意味着可以宽宏大量地允许那些可能没有（13.6）所做得那样完美的浮点计算的硬件实现．对于这样的机器，如果 $\epsilon_{机器}$ 代之以某个大些的值，（13.5）和（13.7）仍然满足．例如，在计算机上中间量被截尾而非舍入，当 $\epsilon_{机器}$ 换成 $2\epsilon_{机器}$ 时，（13.7）仍然成立．

解决如此复杂情况的最简单的方法是保持（13.5）和（13.7）的写法，但是修改 $\epsilon_{机器}$ 的定义．从现在开始，假设 $\epsilon_{机器}$ 不再由（13.3）定义，而是作为使（13.5）和（13.7）成立的最小的数．对大多数计算机，包含所有实现 IEEE 运算的那些机器，$\epsilon_{机器}$ 定义的这个改变在它的价值方面并没有实质的变化．

（13.7）的成立偶尔也需要一个意外大的 $\epsilon_{机器}$ 值．在 1994 年晚期，Intel Pentium™ 微处理器得到了一个恶名，这是因为在实现双精度 IEEE 标准的一个表中出现了一个缺陷，它的有效精度是十分粗糙的 11 阶数量级，$\epsilon_{机器} \approx 6.1 \times 10^{-5}$.（这个缺陷立刻得到改正．）事实上，仍存在一些机器，只对 $\epsilon_{机器} = 1$（13.7）才成立．例如，直到 20 世纪 90 年代中期还在生产的 Cray 计算机上的浮点减法就有这个性质，因为减法的实现没有“保护位”．这样的计算机并不是没有用的，但是它们需要有一种不同于本书中的误差分析方式．

幸而，公理（13.7）和采用计算机运算统一标准的好处在最近几年被计算机制造厂家们广泛地接受，市场上那些需要较大的 $\epsilon_{机器}$ 值（13.7）式才成立的计算机数目正在减少．的确，IEEE 运算已快速地成为所有大小计算机的标准，这包括了（从 1996 年起）所有 IBM 兼容个人计算机和所有由 SUN，DEC，HP 和 IBM 生产的工作站．

13.6 复浮点运算

浮点复数一般表示为一对浮点实数，在其上的初等运算化作实部和虚部的计算. 结果是公理（13.7）对于复的和实的浮点数同样有效，除了对$\otimes$和$\oplus$，$\epsilon_{机器}$须由（13.3）分别放大$2^{3/2}$和$2^{5/2}$阶的因子以外. 一旦$\epsilon_{机器}$以这种方法修正，对复数的舍入误差分析就可以像对实数那样进行.

100

习 题

13.1 在一对相邻的非零 IEEE 单精度实数之间有多少个 IEEE 双精度数?

13.2 由（13.2）定义的浮点系统 **F** 包含了若干个整数，但不是它们的全部.

(a) 给出不属于 **F** 的最小正整数 n 的一个准确公式.

(b) 尤其是，对 IEEE 单精度和双精度运算的 n 值是什么?

(c) 对你的计算机指出一个验证此结果的方法. 特别设计和运行一个程序，它产生证据说明 $n-3, n-2$ 和 $n-1$ 属于 **F** 而 n 不属于它. $n+1, n+2$ 和 $n+3$ 如何?

13.3 考虑多项式 $p(x) = (x-2)^9 = x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512$.

(a) 对 $x = 1.920, 1.921, 1.922, \cdots, 2.080$ 画出 $p(x)$ 的图，借助于 p 的系数 $1, -18, 144, \cdots$ 计算它的值.

(b) 再次产生同样的图，这次是借助表达式 $(x-2)^9$ 计算 p 的值.

13.4 多项式 $p(x) = x^5 - 2x^4 - 3x^3 + 3x^2 - 2x - 1$ 有 3 个实零点. 对 p 应用初始猜测 $x_0 = 0$ 的牛顿法产生一个估计序列 $x_1, x_2, x_3, \cdots$，它快速收敛到一个零点 $x_* \approx -0.315$.

(a) 在$\epsilon_{机器} \approx 10^{-16}$的浮点运算计算 $x_1, \cdots, x_6$，你估计这些数的每一个中有多少数字位是正确的?

(b) 借助诸如 MAPLE 或 MATHEMATICA 那样的符号代数系统再次准确地计算 $x_1, \cdots, x_6$，每个 x_j 是一个有理数，对每个 j，分子和分母有多少个数字位?

101

第14讲 稳　定　性

如果数值算法能提供数值问题的准确解，那将是一件极好的事情. 然而一般这是不可能的，因为问题是连续的而数字计算机是离散的. 稳定的概念是用来说明“什么是可以允许的”标准途径——取得“正确答案”（甚至它是不准确的）意味什么的数值分析思想.

14.1 算　　法

在第12讲中，一个问题定义为由数据的向量空间 $\boldsymbol{X}$ 到解的向量空间 $\boldsymbol{Y}$ 的一个函数 f: $\boldsymbol{X}\to\boldsymbol{Y}$.

一个算法（algorithm）可以看成两个相同空间之间的另外一个映射 $\tilde{f}:\boldsymbol{X}\to\boldsymbol{Y}$. 下面要使这个定义精确起来. 令一个问题 f，一台浮点系统满足（13.7）的计算机（但不需要满足（13.6）），一个对 f 的算法（在此术语不确切的意义下），以及这个算法以计算机程序形式的一个实现，都加以固定. 给定数据 $\boldsymbol{x}\in\boldsymbol{X}$，令此数据在一个满足（13.5）的情况下舍入到浮点，然后把它们作为输入提供给计算机程序. 现在运行这个程序，结果是一组属于向量空间 $\boldsymbol{Y}$ 的浮点数（因为算法是为解 f 而设计的）. 这些计算结果称为 $\tilde{f}(\boldsymbol{x})$.

情况不应该会太难以处理！最低限度，$\tilde{f}(\boldsymbol{x})$ 会受舍入误差的影响. 依赖于环境，它也可能受其他各种复杂因素的影响，诸如收敛性的允许度或甚至是运行在计算机上的其他作业，在那些直到运行时刻计算任务才被决定分派到处理器的情况. 因此由一个运行到下一个运行，“函数” $\tilde{f}(\boldsymbol{x})$ 甚至可以取不同的值；它可能是多值的.（事实上，问题 f 也实在应该允许多值，它允许处理那些可接受非惟一解的情况，例如一个复数的两个平方根中的一个.）尽管有这些复杂情况，我们仍然要找到那些能够作出关于 $\tilde{f}(\boldsymbol{x})$ 的惊人地清晰的叙述，因此关于数值线性代数算法的准确度，只基于基本公理（13.5）和（13.7）.

102

波形记号（~）是极其常用的，正如 $\tilde{f}$ 是 f 的计算模拟那样，在本书其他的计算量将频繁地用波形记号. 例如，方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的计算解可记为 $\tilde{\boldsymbol{x}}$.

14.2 准　确　度

除了在平凡的情形，$\tilde{f}$ 不能是连续的. 然而，一个好的算法应该逼近所关联的

问题f. 为了使这个思想量化，可以考虑计算的绝对误差$\|\tilde{f}(x)-f(x)\|$，或相对误差

$$\frac{\|\tilde{f}(x)-f(x)\|}{\|f(x)\|}. \tag{14.1}$$

在本书中主要用相对的量，因此（14.1）将是我们标准的误差度量.

如果$\tilde{f}$是一个好的算法，则自然期望相对误差是小的，到$\epsilon_{机器}$阶. 对问题f的一个算法$\tilde{f}$，如果对每个$x\in X$，有

$$\frac{\|\tilde{f}(x)-f(x)\|}{\|f(x)\|}=O(\epsilon_{机器}), \tag{14.2}$$

就可以说算法$\tilde{f}$对问题f是准确的. 不严格地说，（14.2）中的符号$O(\epsilon_{机器})$意味着“在机器ϵ的阶上” 然而$O(\epsilon_{机器})$也有精确的意义，我们将马上讨论. 该讨论中也将阐明像（14.2）那样的公式在分母为零时如何解释.

14.3 稳 定 性

然而，如果问题f是病态的，由（14.2）定义的准确度的目标是不合理的奢望. 在一台数字计算机上，输入数据的舍入是不可避免的，甚至如果所有随后的计算能被完全执行，在结果中这个扰动也可以单独导致有效数字的改变. 在所有情况下，替代准确度 [103] 的意图，最适当的目标是稳定性. 如果对于问题f的一个算法$\tilde{f}$，对于每个$x\in X$，

$$\frac{\|\tilde{f}(x)-f(\tilde{x})\|}{\|f(\tilde{x})\|}=O(\epsilon_{机器}) \tag{14.3}$$

对某些满足

$$\frac{\|\tilde{x}-x\|}{\|x\|}=O(\epsilon_{机器}) \tag{14.4}$$

的$\tilde{x}$成立，就说算法$\tilde{f}$是稳定（stable）的. 用文字说明就是

一个稳定的算法对几乎准确的问题
给出几乎准确的答案.

下这个定义的动机将在下一讲中变得清晰，且在本书的剩余部分中应用.

提醒读者注意，尽管这里给出的稳定性定义在数值线性代数的很多部分是有用的，但对其他领域（如微分方程）所有的数值问题，条件$O(\epsilon_{机器})$可能是太严格的.

14.4 向后稳定性

许多数值线性代数的算法满足一个比稳定性更强又更简单的条件. 对问题f的一个算法$\tilde{f}$，如果对每个$\boldsymbol{x}\in \boldsymbol{X}$,

$$\tilde{f}(\boldsymbol{x}) = f(\tilde{\boldsymbol{x}})\text{对某些满足}\frac{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|} = O(\epsilon_{\text{机器}})\text{的}\tilde{\boldsymbol{x}}\text{成立}, \tag{14.5}$$

就说算法$\tilde{f}$是向后稳定的（backward stable）. 这是稳定性定义的一个加强，在（14.3）中的$O(\epsilon_{\text{机器}})$用零代替. 用文字说明就是

一个向后稳定的算法对几乎准确的问题精确地给出准确的答案.

例子在下一讲中给出.

14.5 $O(\epsilon_{\text{机器}})$的意义

现在解释在（14.2）~（14.5）中“$O(\epsilon_{\text{机器}})$”精确的意义.

记号

$$\varphi(t) = O(\psi(t)) \tag{14.6}$$

是一个有精确意义的标准数学记号。此方程指出存在某个正数C，使得对所有充分接近于一个明确极限的t（例如$t\to 0$或$t\to\infty$），有

104

$$|\varphi(t)| \leqslant C\psi(t). \tag{14.7}$$

例如，当$t\to 0$时，$\sin^2 t = O(t^2)$的陈述指出存在常数C，使得对充分小的t，有$|\sin^2 t|\leqslant Ct^2$.

在数学中下面形式的陈述也是标准的：

$$\varphi(s,t) = O(\psi(t)) \quad \text{对}\ s\ \text{一致成立}, \tag{14.8}$$

其中φ是不仅依赖于t，同时也依赖于另一变量s的函数. “一致”这个词指出如同在（14.7）中那样，存在单个的常数C，使对所有s的选择都成立. 例如

$$(\sin^2 t)(\sin^2 s) = O(t^2)$$

当$t\to 0$时一致地成立，但如果用s^2代替$\sin^2 s$就丧失了一致性.

在本书中，按这些标准的定义使用符号“O”. 特别地，我们常常按照

$$\|\text{计算的量}\| = O(\epsilon_{\text{机器}}) \tag{14.9}$$

的方式陈述我们的结果. 这里来说明（14.9）是什么意思. 第一，“$\|$计算的量$\|$”

表示某个数或数集的范数，这些数取决于对问题 f 的算法 $\tilde{f}$，同时也依赖于 f 的数据 $x\in X$ 和依赖于 $\epsilon_{机器}$，一个例子就是相对误差（14.1）. 第二，隐含的极限过程是 $\epsilon_{机器}\to 0$（即 $\epsilon_{机器}$ 是（14.8）中对应 t 的变量），第三，"O"一致地对所有数据 $x\in X$ 使用（即 x 是对应于 s 的变量）. 我们将很少指出对于 $x\in X$ 的一致性，但它总是隐含的.

在任何特定的机器运算中，数 $\epsilon_{机器}$ 是一个固定的量. 在说到极限 $\epsilon_{机器}\to 0$ 时，我们考虑一个理想化的计算机，或者可以说一个计算机家族的理想化. 方程（14.9）意味着，如果我们在满足（13.5）和（13.7）的计算机上，对递减到零的 $\epsilon_{机器}$ 值的序列运行问题的算法，则 $\|$计算的量$\|$ 将保证按 $\epsilon_{机器}$ 的比例减少，或减少得更快. 这些理想计算机需要满足（13.5）和（13.7）而没有其他要求.

14.6 依赖于 m 和 n，不依赖 A 和 b

不妨更深入一点讨论（14.2）~（12.5）中的 $O(\epsilon_{机器})$ 的意义。隐含在"O"中的常数的一致性可以用下面例子说明. 假设考虑对 x 解非奇异 $m\times m$ 方程组 $Ax=b$ 的一个算法，我们宣称对此算法的计算结果 $\tilde{x}$ 满足

$$\frac{\|\tilde{x}-x\|}{\|x\|}=O(\kappa(A)\epsilon_{机器}). \qquad (14.10)$$

[105]

这个叙述意味着界

$$\frac{\|\tilde{x}-x\|}{\|x\|}\leqslant C\kappa(A)\epsilon_{机器} \qquad (14.11)$$

对单独的常数 C 成立，对所有充分小的 $\epsilon_{机器}$，C 独立于矩阵 A 或右端向量 b.

如果像（14.11）那样的公式中分母为零，它的意义由下面的习惯确定. 当写出（14.11）时，我们真实的意思是

$$\|\tilde{x}-x\|\leqslant C\kappa(A)\epsilon_{机器}\|x\|. \qquad (14.12)$$

这在 $\|x\|\neq 0$ 时没有什么不同，但如果 $\|x\|=0$，（14.12）就清楚地表明了（14.10）的精确意义是对所有充分小的 $\epsilon_{机器}$ 有 $\|\tilde{x}-x\|=0$.

虽然（14.11）和（14.12）中的常数 C 不依赖于 A 和 b，但一般它依赖于维数 m. 形式上，这是在第 12 讲中关于问题的定义的一个推论. 如果定义在问题 f 的 m 或 n 那样的维数发生改变，那么向量空间 X 和 Y 也一定要改变，因此我们有一个新的问题，f'. 实际上数值线性代数算法的舍入误差效果一般也随 m 和 n 增长. 然而，这种增长通常是足够慢的以致于情况并不严重. 对于 m 和 n 的依赖性的典型是线性，二次，或最坏情形是三次的（指数既依赖于范数的选择，又依赖于算法的选择），且

大多数数据的误差一定大大地小于最坏的情形，这是因为有统计相消的原因.

原则上，(14.9) 那样的陈述可能隐藏了一个如 2^m 那样的维数 - 依赖因子，它会造成一个在实际上是无用的界. 然而，这样的事情在本书中只在一处发生——在部分选主元的高斯消元法的讨论中——我们将在这点给读者充分的警告以避免误解. 作为一个规则，在本书中当出现 $O(\epsilon_{机器})$ 的时候，希望在一台实际的机器上的实际计算中，问题中的量至多是 100 或 1000 倍 $\epsilon_{机器}$ 的大小.

14.7　范数的独立性

包括 $O(\epsilon_{机器})$ 我们的定义有一个方便的性质；假如 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 是有限维的，那它们与范数无关.

定理 14.1　对于定义在有限维空间 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 上的问题 f 和算法 $\tilde{f}$，准确度，稳定性和向后稳定性的性质成立或失效都与对 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 中范数的选择无关.

106 **证明**　众所周知（且易证明），在有限维向量空间所有范数等价，其意义是若 $\|\cdot\|$ 和 $\|\cdot\|'$ 是同一个空间上的两种范数，则存在正常数 C_1 和 C_2，使得 $C_1\|\boldsymbol{x}\| \leq \|\boldsymbol{x}\|' \leq C_2\|\boldsymbol{x}\|$ 对空间中所有的 $\boldsymbol{x}$ 成立. 由此可得，范数的改变影响常数 C 的大小，此常数隐含在包括 $O(\epsilon_{机器})$ 的陈述中，而不影响这样的常数的存在性. □

习　题

14.1　判断真或假.

(a) 当 $x\to\infty$ 时 $\sin x = O(1)$.

(b) 当 $x\to 0$ 时 $\sin x = O(1)$.

(c) 当 $x\to\infty$ 时 $\log x = O(x^{1/100})$.

(d) 当 $n\to\infty$ 时 $n! = O((n/e)^n)$.

(e) 当 $V\to\infty$ 时 $A = O(V^{2/3})$，其中 A 和 V 是分别用平方微米和立方英里度量的一个球的曲面面积和体积.

(f) $\mathrm{fl}(\pi) - \pi = O(\epsilon_{机器})$.（没有指出极限 $\epsilon_{机器}\to 0$，因为这隐含在本书所有的 $O(\epsilon_{机器})$ 的表达式中).

(g) $\mathrm{fl}(n\pi) - n\pi = O(\epsilon_{机器})$，对所有整数一致地成立.（这里 $n\pi$ 表示准确的数学量，不是浮点计算的结果.）

14.2　(a) 证明 $(1+O(\epsilon_{机器}))(1+O(\epsilon_{机器})) = 1+O(\epsilon_{机器})$. 这个陈述精确的意义为，如果 f 是一个函数，当 $\epsilon_{机器}\to 0$ 时，满足 $f(\epsilon_{机器}) = (1+O(\epsilon_{机器}))(1+O(\epsilon_{机器}))$，则当 $\epsilon_{机器}\to 0$ 时 f 也满足 $f(\epsilon_{机器}) = 1+O(\epsilon_{机器})$.

107 (b) 证明 $(1+O(\epsilon_{机器}))^{-1} = 1+O(\epsilon_{机器})$.

第 15 讲　稳定性的进一步讨论

我们通过稳定和不稳定算法的例子来继续讨论稳定性. 此外我们还讨论联系条件和稳定性的一个基本思想，即向后误差分析，自 20 世纪 50 年代以来，它的威力已经在无数的应用中得到证明.

15.1　浮点运算的稳定性

4 个最简单的计算问题是 $+$，$-$，$\times$ 和 $\div$. 关于算法的选择没有什么可说的了. 的确，常规上，我们仍将使用计算机提供的浮点运算 $\oplus$，$\ominus$，$\otimes$ 和 $\oslash$. 如果这样做，那么公理（13.5）和（13.7）隐含了以下这 4 个算法的标准例子都是向后稳定的.

我们对减法显示这个性质，因为这在初等运算中，减法最有可能被预期为不稳定. 如在例 12.3 中，数据空间 $\boldsymbol{X}$ 是 2 维向量集合 $\mathbb{C}^2$，解空间 $\boldsymbol{Y}$ 是数量集合 $\mathbb{C}$，由定理 14.1，不需要特别指定这些空间的范数.

对于数据 $\boldsymbol{x}=(x_1,x_2)^*\in\boldsymbol{X}$，减法问题对应于函数 $f(x_1,x_2)=x_1-x_2$，我们考虑的算法可以写成

$$\tilde{f}(x_1,x_2)=\mathrm{fl}(x_1)\ominus\mathrm{fl}(x_2).$$

此方程意味着首先将 x_1 和 x_2 舍入为浮点值，然后应用运算 $\ominus$. 现由（13.5），有

$$\mathrm{fl}(x_1)=x_1(1+\epsilon_1),\qquad \mathrm{fl}(x_2)=x_2(1+\epsilon_2)$$

108

对某些 $|\epsilon_1|,|\epsilon_2|\leqslant\epsilon_{\text{机器}}$ 成立. 由（13.7），有

$$\mathrm{fl}(x_1)\ominus\mathrm{fl}(x_2)=(\mathrm{fl}(x_1)-\mathrm{fl}(x_2))(1+\epsilon_3)$$

对某个 $|\epsilon_3|\leqslant\epsilon_{\text{机器}}$ 成立. 组合这些方程给出

$$\begin{aligned}\mathrm{fl}(x_1)\ominus\mathrm{fl}(x_2)&=[x_1(1+\epsilon_1)-x_2(1+\epsilon_2)](1+\epsilon_3)\\&=x_1(1+\epsilon_1)(1+\epsilon_3)-x_2(1+\epsilon_2)(1+\epsilon_3)\\&=x_1(1+\epsilon_4)-x_2(1+\epsilon_5)\end{aligned}$$

对某些 $|\epsilon_4|,|\epsilon_5|\leqslant 2\,\epsilon_{\text{机器}}+O(\epsilon_{\text{机器}}^2)$ 成立（参看习题 14.2）. 换句话说，计算结果 $\tilde{f}(x)=\mathrm{fl}(x_1)\ominus\mathrm{fl}(x_2)$ 正好等于差 $\tilde{x}_1-\tilde{x}_2$，其中 $\tilde{x}_1$ 和 $\tilde{x}_2$ 满足

$$\frac{|\tilde{x}_1-x_1|}{|x_1|}=O(\epsilon_{\text{机器}}),\qquad \frac{|\tilde{x}_2-x_2|}{|x_2|}=O(\epsilon_{\text{机器}}),$$

并且任意的 $C>2$ 将足以满足隐含在符号“O”中的常数. 对于空间 $\mathbb{C}^2$ 中范数

$\|\cdot\|$的任何选择，它隐含了 (14.5).

15.2 进一步的例子

例 15.1 内积 假设给定向量 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{C}^m$，希望计算内积 $\alpha=\boldsymbol{x}^*\boldsymbol{y}$. 显而易见的算法是，用⊗计算两两的乘积 $\bar{x}_i y_i$，用⊕将它们加起来得到计算结果 $\tilde{\alpha}$. 可以证明这个算法是向后稳定的，这隐含在第 17 讲中. □

例 15.2 外积 另一方面，假设对向量 $\boldsymbol{x}\in\mathbb{C}^m,\boldsymbol{y}\in\mathbb{C}^n$ 要计算秩 1 外积 $\boldsymbol{A}=\boldsymbol{x}\boldsymbol{y}^*$，显而易见的算法是，用⊗计算 mn 个乘积 $x_i\bar{y}_j$，并将它们收集到矩阵 $\tilde{\boldsymbol{A}}$ 中. 这个算法是稳定的，但不向后稳定. 对此的解释是，矩阵 $\tilde{\boldsymbol{A}}$ 最不可能具有秩 1，所以它一般不能写成 $(\boldsymbol{x}+\delta\boldsymbol{x})(\boldsymbol{y}+\delta\boldsymbol{y})^*$ 的形式. 作为一个法则，对于那些解空间 $\boldsymbol{Y}$ 的维数大于问题空间 $\boldsymbol{X}$ 维数的问题，很少有向后稳定性成立的. □

例 15.3 假设用⊕计算 $x+1$，给出 $x\in\mathbb{C}$，$\tilde{f}(x)=\mathrm{fl}(x)\oplus 1$. 这个算法是稳定的但不是向后稳定的. 理由是，对 $x\approx 0$，加法⊕将导致如 $O(\epsilon_{机器})$ 大小的绝对误差. 相对于 x 的大小，这是无界的，所以它们不能解释为由在数据中小的相对扰动所引起的. 这个例子指出向后稳定性是更加特别的性质，在某些情况中是一个合理的目标，但在其他情况则不是. 注意，如果问题对数据 x 和 y 已经计算了 $x+y$，则算法已经是向后稳定的. □

109

例 15.4 对于计算 $\sin x$ 或 $\cos x$ 的计算机程序或计算器合理的期望是什么？答案再次是稳定但不是向后稳定的. 对于 $\cos x$，这可由 $\cos 0\neq 0$ 的事实得出，如在上例中那样. 对于 $\sin x$ 和 $\cos x$ 两者，向后稳定性也由函数在某些点的导数为零这个事实所排除. 例如，假设在计算机上对 $x=\pi/2-\delta$，$0<\delta\ll 1$ 计算 $f(x)=\sin x$ 的值，假设我们足够幸运地得到了作为计算结果的极其正确的答案，舍入到浮点数系统：$\tilde{f}(x)=\mathrm{fl}(\sin x)$. 因为 $f'(x)=\cos x\approx\delta$，对某些满足 $\tilde{x}-x\approx(\tilde{f}(x)-f(x))/\delta=O(\epsilon_{机器}/\delta)$ 的 $\tilde{x}$ 有 $\tilde{f}(x)=f(\tilde{x})$. 因为 δ 可以任意小，所以这个向后误差不是 $O(\epsilon_{机器})$ 数量级. □

15.3 一个不稳定的算法

上述都是些教学例子. 此处是一个更有实际意义的例子：用特征多项式去求矩阵的特征根.

因 z 是 $\boldsymbol{A}$ 的特征根当且仅当 $p(z)=0$，其中 $p(z)$ 是特征多项式 $\det(z\boldsymbol{I}-\boldsymbol{A})$，$p$ 的根就是 $\boldsymbol{A}$ 的特征根（第 24 讲）. 这给出了一个计算特征根的方法：

1. 求特征多项式的系数.
2. 求它的根.

这个算法不仅不向后稳定，也是不稳定的，所以不能应用. 甚至在选取特征根是良态问题的情况下，可能产生相对误差极大地大于$\epsilon_{机器}$的答案.

不稳定性在第 2 步求根中展现出来. 如同在例 12.5 中所见的，给出多项式系数求它的根的问题一般是病态的. 随之而来的是特征多项式系数小的误差将导致求根时误差被放大，即使求根做得完全准确.

例如，设 $\boldsymbol{A}=\boldsymbol{I}$ 为 2×2 单位矩阵. $\boldsymbol{A}$ 的特征根对元素的扰动是不灵敏的，一个稳定的算法带着误差 $O(\epsilon_{机器})$ 应该能够计算它们. 然而，上面描述的算法产生 $\sqrt{\epsilon_{机器}}$阶的误差. 为了解释这一点，注意特征多项式是 x^2-2x+1，和例 12.5 中的一样. 当计算多项式的系数时，它们会有$\epsilon_{机器}$阶的误差，这会引起根在 $\sqrt{\epsilon_{机器}}$阶的改变. 例如，若$\epsilon_{机器}=10^{-16}$，则被计算的特征多项式的根可以有由实际特征值近乎 10^{-8}的扰动，这损失了 8 位数字的准确度.

在你尝试自己做这个计算之前，我们必须给出进一步的忠告. 如果用刚才描述的方法计算 2×2 单位矩阵的特征根，你可能会发现根本没有误差，因为 x^2-2x+1 的系数和根是小的整数，它在你的计算机上将会准确地表示. 但是如果是在一个轻微扰动的矩阵，如在

110

$$\boldsymbol{A}=\begin{bmatrix}1+10^{-14} & 0\\ 0 & 1\end{bmatrix},$$

上做实验，计算的特征值将会不同于实际的特征值，它有所期望的 $\sqrt{\epsilon_{机器}}$阶. 试试看!

15.4　向后稳定算法的准确度

假设对问题f: $\boldsymbol{X}\to\boldsymbol{Y}$ 有一个向后稳定的算法 $\tilde{f}$. 它给出的结果准确吗? 答案依赖于f 的条件数 $\kappa=\kappa(\boldsymbol{x})$. 如果 $\kappa(\boldsymbol{x})$小，那么结果在相对的意义下是准确的，但若它大，则准确度会成比例地受到损失.

定理 15.1　*假设一个向后稳定的算法在一个满足公理（13.5）和（13.7）的计算机上解具有条件数 κ 的问题f:$\boldsymbol{X}\to\boldsymbol{Y}$，则相对误差满足*

$$\frac{\|\tilde{f}(\boldsymbol{x})-f(\boldsymbol{x})\|}{\|f(\boldsymbol{x})\|}=O(\kappa(\boldsymbol{x})\epsilon_{机器}). \tag{15.1}$$

证明　由向后稳定性的定义（14.5），对某些满足

$$\frac{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|}=O(\epsilon_{机器})$$

的$\tilde{x}\in X$可得$\tilde{f}(x)=f(\tilde{x})$. 由$\kappa(x)$的定义（12.5），这隐含了

$$\frac{\|\tilde{f}(x)-f(x)\|}{\|f(x)\|}\leqslant(\kappa(x)+o(1))\frac{\|\tilde{x}-x\|}{\|x\|},\tag{15.2}$$

其中$o(1)$记为当$\epsilon_{机器}\to 0$时收敛到零的一个量. 组合这些界便给出（15.1）. □

15.5 向后误差分析

刚才在证明定理15.1中的过程称为*向后误差分析*（backward error analysis）. 我们通过两步得到准确度的估计. 一步是研究问题的条件，另一步是研究算法的稳定性. 概括地说，如果算法是稳定的，则最后的准确度反映了条件数的效果.

111

从数学上看这是明确的，但这不是一个想要分析数值算法而又没有准备好的人会想到的第一个想法. 第一个想法是*向前误差分析*（forward error analysis）. 在这个想法中，每步计算引入的舍入误差被估计，且以某种方式如何将它们一步一步地组合起来，以保持一个总体.

实践已经证明，对数值线性代数大多数的算法来说，向前误差分析比起向后误差分析要更难于实施. 利用事后的认识，不难说明为什么会是这样. 例如说，假设一个可靠准确的算法用在一个计算机上解$Ax=b$，当A是病态时，得到的结果总是有更差的准确度，这是一个确定的事实（参看第22讲）. 现在，向前误差分析怎样捕捉到这个现象？A的条件数是一个整体的性质，以致于它在含于解$Ax=b$的个别的浮点运算水平上或多或少地看不到.（在下一讲中将以一个例子生动地说明这一点.）此外，如果以一个正确的结果结束，向前分析将一定会检测到条件数.

简短地说，这是一个确定的事实，即一般情况下，对大多数问题的最好算法不比对轻微扰动数据计算准确解好. 向后误差分析是一个简洁合理地符合这个向后现实的方法.

习 题

15.1 下列每个问题描述了一个在满足公理（13.5）和（13.7）的计算机上实现的算法. 对每个算法，分析算法是*向后稳定*，*稳定但不向后稳定*，或者是*不稳定的*，并证明你的结论或者给出一个合理的、令人信服的理由. 下述的定义已在正文中给出.

(a) 数据：$x\in\mathbb{C}$. 解：$2x$，用$x\oplus x$计算.

(b) 数据：$x\in\mathbb{C}$. 解：x^2，用$x\otimes x$计算.

(c) 数据：$x\in\mathbb{C}\setminus\{0\}$. 解：1，用$x\oslash x$计算. [满足（13.6）的机器会准确地给出正确的答案，但我们的定义是基于较弱的条件（13.7）.]

(d) 数据：$x\in\mathbb{C}$. 解：0，用$x\ominus x$计算. [再次说明，实际的机器会比基于（13.7）的定义做得更好.]

(e) 数据：没有. 解：e，用和式 $\sum_{k=0}^{\infty} 1/k!$，由左到右用⊗和⊕计算，当被加数达到数量 $<\epsilon_{\text{机器}}$ 时停止.

(f) 数据：没有. 解：e，由上述同样算法计算，只是级数求和由右至左. 112

(g) 数据：没有. 解：π，计算是在区间 [3,4] 中做一个穷举搜索，求满足 $s(x) \otimes s(x') \leqslant 0$ 的最小浮点数 x. 这里 $s(x)$ 是一个在给定区间稳定地计算 $\sin(x)$ 的算法，x' 记为浮点系统中 x 之后的下一个浮点数.

15.2 考虑计算矩阵（完全）SVD 问题的一个算法. 此问题的数据是一个矩阵 $\boldsymbol{A}$，解是满足 $\boldsymbol{A}=\boldsymbol{U}\Sigma \boldsymbol{V}^*$ 的 3 个矩阵 $\boldsymbol{U}$（酉的），Σ（对角的），和 $\boldsymbol{V}$（酉的）.（这里说的是显式的矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$，不是作为镜射算子乘积的隐式表示.）

(a) 说明这个算法向后稳定将意味着什么.

(b) 事实上，因为一个简单的理由，这个算法不能向后稳定. 试作解释.

(c) 幸而，计算 SVD 的标准算法（第 31 讲）是稳定的. 解释这样的算法稳定性意味着什么. 113

第 16 讲　豪斯霍尔德三角形化的稳定性

在本讲中我们在行动上观察向后误差分析．首先在一个 MATLAB 实验中，观察豪斯霍尔德三角形化的向后稳定性的引人注意的性质．然后考虑三角形化的步骤如何能够组合其他向后稳定的部分，来获得解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的一个稳定算法．

16.1 实　　验

豪斯霍尔德因子分解是计算 QR 因子分解的一个向后稳定的算法．我们可以用一个在 IEEE 双精度运算（$\epsilon_{机器}\approx 1.11\times 10^{-16}$）下做的 MATLAB 实验来说明这一点．

代码	说明
`R = triu(randn(50));`	令 $\boldsymbol{R}$ 为正态随机元素的 50×50 上三角矩阵．
`[Q,X] = qr(randn(50));`	令 $\boldsymbol{Q}$ 为由正交化一个随机矩阵得到的随机正交矩阵．
`A = Q * R;`	令 $\boldsymbol{A}$ 为乘积 $\boldsymbol{QR}$，直到舍入误差．
`[Q2,R2] = qr(A);`	由豪斯霍尔德三角形化计算 QR 因子分解 $\boldsymbol{A}\approx\boldsymbol{Q}_2\boldsymbol{R}_2$．

114

这 4 行 MATLAB 程序的目的是构造一个已知 QR 因子分解的矩阵 $\boldsymbol{A}=\boldsymbol{QR}$，然后它可以与由豪斯霍尔德三角形化计算的 QR 因子分解 $\boldsymbol{A}=\boldsymbol{Q}_2\boldsymbol{R}_2$ 比较．实际上，由于存在舍入误差，计算出来的矩阵 $\boldsymbol{A}$ 的 QR 因子并非准确地是 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$．然而，对于这个实验的用意，它们是足够接近的．如果 $\boldsymbol{A}$ 准确地等于 $\boldsymbol{QR}$，所给出的结果没有本质的不同(事实上，通过在计算机上以高精度运算来计算 $\boldsymbol{A}=\boldsymbol{QR}$，我们就可以相信这一点)．

对于 $\boldsymbol{Q}_2$ 和 $\boldsymbol{R}_2$，偏巧它们是离准确非常远的．

代码	说明
`norm (Q2 - Q)`	$\boldsymbol{Q}_2$ 有多准确?
`ans = 0.00889`	
`norm (R2 - R) / norm(R)`	$\boldsymbol{R}_2$ 有多准确?
`ans = 0.00071`	

这些误差是巨大的！我们的计算虽然已经做到 16 位的准确度，但最后的结果仅有 2 或 3 位准确．个体的舍入误差以 10^{13} 阶的因子放大．（注意，计算的 $\boldsymbol{Q}_2$ 足够接近 $\boldsymbol{Q}$ 表明列符号的改变不可能是造成任何误差的原因．如果你尝试这个实验并得到完全不同的结果，可能是需要在 $\boldsymbol{Q}$ 的列和 $\boldsymbol{R}$ 的行上乘以一个适当的因子 ± 1．）

我们似乎失去了 12 位的准确度，但现在当我们乘这些不准确的矩阵 $\boldsymbol{Q}_2$ 和 $\boldsymbol{R}_2$ 时，一件惊人的事情发生了：

`norm (A -Q2 * R2) /norm(A)`	Q_2R_2 有多准确？
`ans =1.432e -15`	

乘积 Q_2R_2 完全准确到 15 位！在 Q_2 和 R_2 中的误差一定是“魔鬼相关”的，这是 Wilkinson 用的说法. 在 10^{12} 中的单独部分，在乘积 Q_2R_2 中抵消了.

为了突出 Q_2R_2 的准确度有怎样的特性，我们构造另一对精确近似 Q 和 R 的矩阵 Q_3 和 R_3，并将它们乘起来.

`Q3 =Q +1e -4 * randn(50);`	令 Q_3 随机扰动 Q，它比 Q_2 更接近 Q.
`R3 =R +1e -4 * randn(50);`	令 R_3 随机扰动 R，它比 R_2 更接近 R.
`norm (A - Q3 * R3) /norm(A)`	Q_3R_3 有多准确？
`ans =0.00088`	

这次，乘积的误差是巨大的. Q_2 并不好于 Q_3，R_2 亦不好于 R_3，但 Q_2R_2 以 12 阶的数量好于 Q_3R_3. 在这个实验中，我们没有在使 R_3 的上三角形和 Q_3 正交方面制造麻烦，但我们这样做时已经有了小小的差别. [115]

在 Q_2 和 R_2 中的误差是*向前误差*（forward error）. 一般地说，大的向前误差可能是一个病态问题或是一个不稳定算法的结果（定理 15.1）. 在我们的实验中，它们应归于前者. 作为一个法则，随机三角形矩阵的列空间序列作为矩阵元素的函数是极端病态的.

在 Q_2R_2 中的误差是*向后误差*（backward error）或*剩余*（residual）. 这些误差的微小提示了豪斯霍尔德三角形化是向后稳定的.

16.2 定　理

事实上，豪斯霍尔德三角形化对所有矩阵 A 及所有满足（13.5）和（13.7）的计算机是向后稳定的. 现在建立一个定理来说明这点，如同本书中大多数稳定性的结论，我们不给出证明.

我们的结论将写成形式

$$\tilde{Q}\tilde{R} = A + \delta A, \tag{16.1}$$

其中 δA 是小的. 即，计算出来的 Q 乘以计算出来的 R 等于给定矩阵 A 的一个小扰动. 然而，一个细小的区别在于这些符号的用法上. 用 $\tilde{R}$，我们指正好所期望的那个矩阵：由豪斯霍尔德三角形化在浮点运算所构成的上三角矩阵. 然而，用 $\tilde{Q}$，我们指*准确地是酉的*某个矩阵. 重温 Q 等于乘积 $Q = Q_1Q_2\cdots Q_n$（10.7），其中 Q_k 是由

在算法 10.1 的第 k 步确定的向量 v_k（10.4）所定义的豪斯霍尔德镜射算子. 在浮点计算中，我们得到替代向量 $\tilde{v}_k$ 的序列. 令 $\tilde{Q}_k$ 记为由浮点向量 $\tilde{v}_k$ 定义的准确地是酉的镜射算子——是数学的，而非计算机上的. 现在定义

$$\tilde{Q} = \tilde{Q}_1 \tilde{Q}_2 \cdots \tilde{Q}_n. \tag{16.2}$$

准确的酉矩阵 $\tilde{Q}$ 将担当"计算的 Q"的角色. 这个定义初看起来似乎很奇怪，但它却是自然的. 在应用方面，如在第 10 讲中讨论的，矩阵 Q 一般不以任何方式形成显式，所以定义一个更明显类型的"计算的 Q"是没有用的. 向量 $\tilde{v}_k$ 是显式地形成的，这些是（16.2）包含的内容.

下面是说明我们的 MATLAB 实验的定理.

116 **定理 16.1**　令矩阵 $A \in \mathbb{C}^{m\times n}$ 的 QR 因子分解是由豪斯霍尔德三角形化（算法 10.1）在一台满足公理（13.5）和（13.7）的机器上计算的，且令计算的因子 $\tilde{Q}$ 和 $\tilde{R}$ 如上所指地定义. 则有

$$\tilde{Q}\tilde{R} = A + \delta A, \qquad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{机器}}) \tag{16.3}$$

对某个 $\delta A \in \mathbb{C}^{m\times n}$ 成立.

一如此书的惯例，(16.3) 表示式中的 $O(\epsilon_{\text{机器}})$ 有其在第 14 讲中所讨论的精确意义，其界当 $\epsilon_{\text{机器}} \to 0$ 时对所有任意固定的维数 m 和 n 的矩阵 A 一致地成立，但不一致地对 m 和 n 成立. 因为在一个有限维向量空间中所有范数等价，所以我们不需要指定范数（定理 14.1）.

16.3　分析解 $Ax=b$ 的一个算法

我们已经看到，豪斯霍尔德三角形化是向后稳定的但按向前意义并不总是准确的.（这对数值线性代数中大多数的矩阵因子分解同样是如此.）现在，QR 因子分解一般地不在它自身终止，而是到诸如解方程组、最小二乘问题或特征值问题等其他最终目标的一个中间步骤终止. 它的向后稳定性是否足以使它是一个更大算法中令人满意的一段？也就是，乘积 QR 的准确度对于应用来说是否足够，或者是否需要 Q 和 R 分别的准确度？

令人愉快的答案是，实际上 QR 的准确度确实对于大多情况来说是足够的. 我们可用十分简单的理由来说明这一点.

考虑一个用豪斯霍尔德三角形化求解一个非奇异的 $m\times m$ 线性方程组 $Ax=b$ 的例子. 这个思想曾在第 7 讲的末尾讨论过，这里是那个算法更完备的陈述.

算法 16.1 用 QR 因子分解解 $Ax=b$

$QR=A$ 用算法 10.1 分解 A 为 QR，其中 Q 表示为镜射算子的乘积.
$y=Q^*b$ 用算法 10.2 构造 Q^*b.
$x=R^{-1}y$ 用回代（算法 17.1）解三角形方程组 $Rx=y$.

这个算法是向后稳定的，它的证明是直接的，所给出的 3 步中的每一步本身都向后稳定. 这里不加证明地给出这 3 步向后稳定的结论，然后给出它们如何组合的详细处理.

算法 16.1 的第 1 步是 A 的 QR 因子分解，计算矩阵 $\tilde{R}$ 和 $\tilde{Q}$. 这个过程的向后稳定性已在（16.3）中显示.

第 2 步是用算法 10.2 计算 $\tilde{Q}^*b$.（注意，为了这个论述的目的，我们不写 Q^*b，因为在此处的计算，第 1 步是完备的，已经产生的矩阵不是 Q 而是 $\tilde{Q}$.）当用算法 10.2 计算 $\tilde{Q}^*b$ 时，将造成舍入误差，所以结果将不是准确的 $\tilde{Q}^*b$. 取而代之的是某个向量 $\tilde{y}$，可以证明这个向量满足下列向后稳定估计：

117

$$(\tilde{Q}+\delta Q)\tilde{y}=b,\qquad \|\delta Q\|=O(\epsilon_{机器}) \tag{16.4}$$

像（16.3）那样，这个等式是准确的. 用文字说就是，在浮点运算中应用豪斯霍尔德镜射算子的结果准确地等价于以一个轻微扰动矩阵 $(\tilde{Q}+\delta Q)^{-1}$乘以 b.

算法 16.1 的最后一步是回代计算 $\tilde{R}^{-1}\tilde{y}$.（同样，在这个阶段是现有的 $\tilde{R}$ 和 $\tilde{y}$，R 和 y 对它没有作用.）在这步将引入新的舍入误差，但再次计算是向后稳定的. 这时估计取形式

$$(\tilde{R}+\delta R)\tilde{x}=\tilde{y},\qquad \frac{\|\delta R\|}{\|\tilde{R}\|}=O(\epsilon_{机器}) \tag{16.5}$$

如通常那样，左边的等式是准确的. 它断言浮点结果 $\tilde{x}$ 是方程组 $\tilde{R}x=\tilde{y}$ 的一个轻微扰动的准确解.

在下一讲中将完全详尽地导出（16.5），这比起它的表述更令人感兴趣，这个结果陈述为定理 17.1.

现在，假定（16.3）~（16.5）已知，下面是其得到的定理. 这是一个对若干数值线性代数算法可以证明的向后稳定定理的典型.

定理 16.2 算法 16.1 是向后稳定的，对某个 $\Delta A\in\mathbb{C}^{m\times m}$ 满足

$$(A+\Delta A)\tilde{x}=b,\qquad \frac{\|\Delta A\|}{\|A\|}=O(\epsilon_{机器}) \tag{16.6}$$

证明　综合（16.4）和（16.5），有

$$b=(\tilde{Q}+\delta Q)(\tilde{R}+\delta R)\tilde{x}=[\tilde{Q}\tilde{R}+(\delta Q)\tilde{R}+\tilde{Q}(\delta R)+(\delta Q)(\delta R)]\tilde{x}.$$

因而，由（16.3）有

$$b=[A+\delta A+(\delta Q)\tilde{R}+\tilde{Q}(\delta R)+(\delta Q)(\delta R)]\tilde{x}.$$

此方程有

$$b=(A+\Delta A)\tilde{x},$$

的形式，其中 ΔA 是4项的和. 为了建立（16.6），应证明这些项的每一项相对于 A 来说是小的.

118

因 $\tilde{Q}\tilde{R}=A+\delta A$ 且 $\tilde{Q}$ 是酉的，由（16.3）有

$$\frac{\|\tilde{R}\|}{\|A\|}\leqslant\|\tilde{Q}^*\|\frac{\|A+\delta A\|}{\|A\|}=O(1)$$

当 $\epsilon_{机器}\to 0$ 时成立.（如果 $\|\cdot\|=\|\cdot\|_2$，它是 $1+O(\epsilon_{机器})$，但关于一般的 $\|\cdot\|$ 我们没有作假设.）这由（16.4）给出

$$\frac{\|(\delta Q)\tilde{R}\|}{\|A\|}\leqslant\|\delta Q\|\frac{\|\tilde{R}\|}{\|A\|}=O(\epsilon_{机器}).$$

类似地由（16.5），有

$$\frac{\|\tilde{Q}(\delta R)\|}{\|A\|}\leqslant\|\tilde{Q}\|\frac{\|\delta R\|\,\|\tilde{R}\|}{\|\tilde{R}\|\,\|A\|}=O(\epsilon_{机器}).$$

最后，

$$\frac{\|(\delta Q)(\delta R)\|}{\|A\|}\leqslant\|\delta Q\|\frac{\|\delta R\|}{\|A\|}=O(\epsilon_{机器}^2).$$

因此全部的扰动 ΔA 满足

$$\frac{\|\Delta A\|}{\|A\|}\leqslant\frac{\|\delta A\|}{\|A\|}+\frac{\|(\delta Q)\tilde{R}\|}{\|A\|}+\frac{\|\tilde{Q}(\delta R)\|}{\|A\|}+\frac{\|(\delta Q)(\delta R)\|}{\|A\|}=O(\epsilon_{机器}),$$

正如所要求的那样.　□

组合定理12.2，15.1和16.2，给出下面关于 $Ax=b$ 解的准确度的结论：

定理 16.3　由算法16.1计算的解 $\tilde{x}$ 满足

$$\frac{\|\tilde{x}-x\|}{\|x\|}=O(\kappa(A)\epsilon_{机器}) \tag{16.7}$$

习　题

16.1 (a) 令 $\boldsymbol{Q}_1,\cdots,\boldsymbol{Q}_k\in\mathbb{C}^{m\times m}$为固定的酉矩阵，考虑对 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$，乘积 $\boldsymbol{B}=\boldsymbol{Q}_k\cdots\boldsymbol{Q}_1\boldsymbol{A}$ 的计算问题. 令计算过程是由右到左在满足（13.5）和（13.7）的计算机上作直接的浮点运算. 试证明这个算法是向后稳定的.（这里 $\boldsymbol{A}$ 作为数据可以被扰动. 矩阵 $\boldsymbol{Q}_j$ 是固定的，没有扰动.） 119

(b) 给出一个例子以示若酉矩阵 $\boldsymbol{Q}_j$ 用任意矩阵 $\boldsymbol{X}_j\in\mathbb{C}^{m\times m}$代替，则上述结论不再成立.

16.2 本题的想法是做一个模拟本讲所描述的一个实验，但是用 SVD 替代 QR 因子分解.

(a) 写出一个 MATLAB 程序，构造一个 50×50 矩阵`A = U * S * V'`，其中 `U` 和 `V` 是随机正交矩阵，`S` 是对角矩阵，其对角元素为 [0,1] 中的随机同分布的数，整理成不增的次序. 用你的程序计算 `[U2,S2,V2] = svd(A)` 及 `U - U2`，`V - V2`，`S - S2` 和 `A - U2 * S2 * V2'`的范数，对 5 个矩阵 $\boldsymbol{A}$ 做以上工作并评述结果.［提示：`diag(U2' * U)` 和 `diag(V2' * V)` 的图会提供信息.］

(b) 在你计算的 SVD 中固定其符号使得（a）的困难得以克服. 再次对 5 个随机矩阵以各种范数运行程序并加以评述. 它们和 `cond(A)` 有联系吗?

(c) 将 `S` 的对角元素以它们的 6 次幂代之，并重复（b）. 你看到本题的结论和 QR 因子分解实验之间有本质不同吗? 120

第 17 讲　回代的稳定性

数值线性代数最容易的问题之一是解三角形方程组. 标准的方法是逐次代入，当方程组是上三角形时称为回代. 这里我们充分详细地显示这个算法是向后稳定的，得出舍入误差效果的数量界，不带有 "$O(\epsilon_{机器})$".

17.1　三角方程组

我们已经见过，一般的方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 可以用 QR 因子分解化为一个上三角方程组 $\boldsymbol{Rx}=\boldsymbol{y}$. 下三角形方程组和上三角形方程组也出现在高斯消元法中，楚列斯基因子分解及众多的其他数值线性代数计算中. 这些方程组容易用逐次代入过程求解，如果方程组是下三角形的，称为向前代入 (forward substitution)，如果是上三角形的，称为回代 (back substitution). 尽管这两种情形在数学上是等价的，但为了确定，在本讲中讨论回代.

假设要解 $\boldsymbol{Rx}=\boldsymbol{b}$，即

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ & r_{22} & & \vdots \\ & & \ddots & \\ & & & r_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \tag{17.1}$$

其中给定 $\boldsymbol{b}\in\mathbb{C}^m$，非奇异上三角形矩阵 $\boldsymbol{R}\in\mathbb{C}^{m\times m}$，而 $\boldsymbol{x}\in\mathbb{C}^m$ 是未知的. 我们可以对 $\boldsymbol{x}$ 的分量一个接着另一个地求解，由 x_m 开始到 x_1 结束. 为了后面方便，将算法写成一系列的公式而不是一个循环.

算法 17.1　回代

$$x_m = b_m / r_{mm}$$

$$x_{m-1} = (b_{m-1} - x_m r_{m-1,m}) / r_{m-1,m-1}$$

$$x_{m-2} = (b_{m-2} - x_{m-1} r_{m-2,m-1} - x_m r_{m-2,m}) / r_{m-2,m-2}$$

$$\vdots$$

$$x_j = \Big(b_j - \sum_{k=j+1}^{m} x_k r_{jk}\Big) \Big/ r_{jj}$$

结构是三角形的，在每个位置有一次减法一次乘法. 运算计数因而是 $m\times m$ 三角形面积的 2 倍:

$$\text{回代的工作量：} \sim m^2 \text{ 次 flop.} \tag{17.2}$$

17.2　向后稳定定理

在前一讲，回代作为用 QR 因子分解解 $Ax=b$ 的三步中的一步出现. 在（16.3）~（16.5）中，我们断言三步的每一步都是向后稳定的，但没有证明这些论断. 在本讲将用推导含于（16.5）中的界来完成其中的一个证明. 这是一个如何证明向后稳定性被组织起来的例子，边是本书中给出详细证明的惟一情形.

然而，在能够证明算法 17.1 向后稳定之前，必须要对算法的一个细节加以约束，这个约束并没有在所写的公式中特别规定. 在上面圆括号内的表示式中，我们随意地约定，减法是由左到右进行的.（其他的次序也是稳定的，只是估计的细节有所不同.）现在可以陈述我们的定理了.

定理 17.1　令算法 17.1 应用到由在一台满足式子（13.7）的计算机上的浮点数组成的问题（17.1）. 这个算法是向后稳定的，其意义是计算的解 $\tilde{x}\in\mathbb{C}^m$ 满足

$$(R+\delta R)\tilde{x} = b \tag{17.3}$$

122

对某个具有性质

$$\frac{\|\delta R\|}{\|R\|} = O(\epsilon_{\text{机器}}) \tag{17.4}$$

的上三角形矩阵 $\delta R\in\mathbb{C}^{m\times m}$ 成立. 特别地，对每个 i,j，有

$$\frac{|\delta r_{ij}|}{|r_{ij}|} \leqslant m\,\epsilon_{\text{机器}} + O(\epsilon_{\text{机器}}^2). \tag{17.5}$$

在（17.5）以及贯穿本讲的内容中，我们继续用（14.12）的惯例，即如果分母为零，分子也隐含规定为零（对所有充分小的 $\epsilon_{\text{机器}}$）.

为了保持概念的清晰和趣味性，我们的证明将尽量轻松一些.

17.3　$m=1$

根据（17.3），我们的任务是将每个浮点误差表示为输入的一个扰动. 让我们从最简单的情形开始，这里 R 是 1×1 的，在此情形回代由单步组成，

$$\tilde{x}_1 = b_1 \oslash r_{11}.$$

（回想第 13 讲中曾指出 $\oslash$，$\otimes$，$\oplus$ 和 $\ominus$ 表示浮点运算.）公理（13.7）对 $\oslash$ 保证了计算的解接近正确：

$$\tilde{x}_1 = \frac{b_1}{r_{11}}(1+\epsilon_1), \qquad |\epsilon_1| \leq \epsilon_{机器}.$$

然而，我们喜欢把误差表示为好像它是由 $\boldsymbol{R}$ 中的扰动所得的结果. 为此目的，令 $\epsilon'_1 = -\epsilon_1/(1+\epsilon_1)$，因此公式变成

$$\tilde{x}_1 = \frac{b_1}{r_{11}(1+\epsilon'_1)}, \qquad |\epsilon'_1| \leq \epsilon_{机器} + O(\epsilon^2_{机器}). \tag{17.6}$$

注意，ϵ'_1 等于 $-\epsilon_1$ 加上ϵ_1^2 阶的一项. 我们可以自由地由分子到分母或反向移动小的相对扰动，结果根据$\epsilon^2_{机器}$阶改变（习题 14.2（b)).

在（17.6）中，等式是准确的；除法是数学的而不是浮点的. 公式说明 1×1 回代是向后稳定的，因为 $\tilde{x}_1$ 准确地说是一个扰动问题

$$(r_{11}+\delta r_{11})\tilde{x}_1 = b_1$$

的正确解，其中 $\delta r_{11} = \epsilon'_1 r_{11}$. 因而

123

$$\frac{|\delta r_{11}|}{|r_{11}|} \leq \epsilon_{机器} + O(\epsilon^2_{机器}).$$

17.4　$m=2$

2×2 情形稍为少些平凡性. 设有一个上三角形矩阵 $\boldsymbol{R}\in\mathbb{C}^{2\times2}$ 及向量 $\boldsymbol{b}\in\mathbb{C}^2$. $\tilde{\boldsymbol{x}}\in\mathbb{C}^2$的计算分为两步进行. 第一步和 1×1 情形相同：

$$\tilde{x}_2 = b_2 \oplus\!\!\!\!\div r_{22} = \frac{b_2}{r_{22}(1+\epsilon_1)}, \quad |\epsilon_1| \leq \epsilon_{机器} + O(\epsilon^2_{机器}). \tag{17.7}$$

第二步由公式

$$\tilde{x}_1 = (b_1 \ominus (\tilde{x}_2 \otimes r_{12})) \oslash r_{11}.$$

所确定. 为了建立向后稳定性，一定要将这 3 个浮点运算的误差表示为元素 r_{ij} 的扰动.

乘法是容易的，直接用公理（13.7)，解释浮点乘法是在 r_{12}的扰动：

$$\tilde{x}_1 = (b_1 \ominus \tilde{x}_2 r_{12}(1+\epsilon_2)) \oslash r_{11}, \quad |\epsilon_2| \leq \epsilon_{机器}.$$

除法和减法就更微妙了. 首先根据（13.7）写出准确的数学公式：

$$\tilde{x}_1 = (b_1 - \tilde{x}_2 r_{12}(1+\epsilon_2))(1+\epsilon_3) \oslash r_{11} \tag{17.8}$$

$$= \frac{(b_1 - \tilde{x}_2 r_{12}(1+\epsilon_2))(1+\epsilon_3)}{r_{11}}(1+\epsilon_4). \tag{17.9}$$

这里（13.7）保证了 $|\epsilon_3|, |\epsilon_4| \leq \epsilon_{机器}$. 现在将项 ϵ_3 和 ϵ_4 由分子移到分母，如前面那样. 这给出

$$\tilde{x}_1 = \frac{b_1 - \tilde{x}_2 r_{12}(1+\epsilon_2)}{r_{11}(1+\epsilon'_3)(1+\epsilon'_4)},$$

其中 $|\epsilon'_3|, |\epsilon'_4| \leq \epsilon_{机器} + O(\epsilon^2_{机器})$，或等价地

$$\tilde{x}_1 = \frac{b_1 - \tilde{x}_2 r_{12}(1+\epsilon_2)}{r_{11}(1+2\epsilon_5)}, \tag{17.10}$$

其中 $|\epsilon_5| \leq \epsilon_{机器} + O(\epsilon^2_{机器})$. 这公式说明了如果 r_{22}, r_{12} 和 r_{11} 分别以因子（$1+\epsilon_1$），（$1+\epsilon_2$）和（$1+2\epsilon_5$）扰动，那么 $\tilde{x}_1$ 会精确地正确. 这些扰动可以综合为方程

$$(\boldsymbol{R} + \delta\boldsymbol{R})\tilde{\boldsymbol{x}} = \boldsymbol{b},$$

其中 $\delta\boldsymbol{R}$ 的元素 δr_{ij} 满足

$$\begin{bmatrix} |\delta r_{11}|/|r_{11}| & |\delta r_{12}|/|r_{12}| \\ & |\delta r_{22}|/|r_{22}| \end{bmatrix} = \begin{bmatrix} 2|\epsilon_5| & |\epsilon_2| \\ & |\epsilon_1| \end{bmatrix} \leq \begin{bmatrix} 2 & 1 \\ & 1 \end{bmatrix} \epsilon_{机器} + O(\epsilon^2_{机器}).$$

（这里及下面相似结论的“$\leq$”解释为按元素的.）此公式以任意矩阵范数保证了 $\|\delta\boldsymbol{R}\|/\|\boldsymbol{R}\| = O(\epsilon_{机器})$ 成立，因而 2×2 回代是向后稳定的. [124]

17.5　$m=3$

3×3 矩阵的分析包括了对一般情形必须的所有推理. 开始两步和前面相同：

$$\tilde{x}_3 = b_3 \oplus\!\!\!\!\div\; r_{33} = \frac{b_3}{r_{33}(1+\epsilon_1)}, \tag{17.11}$$

$$\tilde{x}_2 = (b_2 \ominus (\tilde{x}_3 \otimes r_{23})) \oplus\!\!\!\!\div\; r_{22} = \frac{b_2 - \tilde{x}_3 r_{23}(1+\epsilon_2)}{r_{22}(1+2\epsilon_3)}, \tag{17.12}$$

其中

$$\begin{bmatrix} 2|\epsilon_3| & |\epsilon_2| \\ & |\epsilon_1| \end{bmatrix} \leq \begin{bmatrix} 2 & 1 \\ & 1 \end{bmatrix} \epsilon_{机器} + O(\epsilon^2_{机器}).$$

第三步计算

$$\tilde{x}_1 = [(b_1 \ominus (\tilde{x}_2 \otimes r_{12})) \ominus (\tilde{x}_3 \otimes r_{13})] \oplus r_{11}. \tag{17.13}$$

通过引入扰动ϵ_4和ϵ_5把（17.13）中的两个$\otimes$转换为数学的乘法：

$$\tilde{x}_1 = [(b_1 \ominus \tilde{x}_2 r_{12}(1+\epsilon_4)) \ominus \tilde{x}_3 r_{13}(1+\epsilon_5)] \oplus r_{11}.$$

借助扰动ϵ_6和ϵ_7把$\ominus$运算转换为数学的减法：

$$\tilde{x}_1 = [(b_1 - \tilde{x}_2 r_{12}(1+\epsilon_4))(1+\epsilon_6) - \tilde{x}_3 r_{13}(1+\epsilon_5)](1+\epsilon_7) \oplus r_{11}.$$

最后，$\oplus$用ϵ_8消去，并立刻用ϵ'_8代之，其中$|\epsilon_8| \leqslant \epsilon_{机器} + O(\epsilon^2_{机器})$，并把结果放到分母：

$$\tilde{x}_1 = \frac{[(b_1 - \tilde{x}_2 r_{12}(1+\epsilon_4))(1+\epsilon_6) - \tilde{x}_3 r_{13}(1+\epsilon_5)](1+\epsilon_7)}{r_{11}(1+\epsilon'_8)}.$$

现在，上述公式除了包含ϵ_6和ϵ_7的项外有我们需要的所有东西，该两项来自于运算$\ominus$. 如果实行分配，将影响到数b_1，而我们的目的是只扰动元素r_{ij}. 包含ϵ_7的项容易迅速了结：改ϵ_7为ϵ'_7并像通常那样把它移到分母. 包含ϵ_6的项需要一个新的窍门，我们也移它到分母，但为了保持等式成立，我们同样把一个新的因子（$1+\epsilon'_6$）放到r_{13}项里来补偿，因此

125

$$\tilde{x}_1 = \frac{b_1 - \tilde{x}_2 r_{12}(1+\epsilon_4) - \tilde{x}_3 r_{13}(1+\epsilon_5)(1+\epsilon'_6)}{r_{11}(1+\epsilon'_6)(1+\epsilon'_7)(1+\epsilon'_8)}.$$

现在r_{13}有2个大小至多为$\epsilon_{机器}$的扰动，r_{11}有3个. 在此公式中，计算中所有的误差都已经表示为$\boldsymbol{R}$的元素的扰动.

结果可综合为

$$(\boldsymbol{R}+\delta\boldsymbol{R})\tilde{\boldsymbol{x}} = \boldsymbol{b},$$

其中元素δr_{ij}满足

$$\begin{bmatrix} |\delta r_{11}|/|r_{11}| & |\delta r_{12}|/|r_{12}| & |\delta r_{13}|/|r_{13}| \\ & |\delta r_{22}|/|r_{22}| & |\delta r_{23}|/|r_{23}| \\ & & |\delta r_{33}|/|r_{33}| \end{bmatrix} \leqslant \begin{bmatrix} 3 & 1 & 2 \\ & 2 & 1 \\ & & 1 \end{bmatrix} \epsilon_{机器} + O(\epsilon^2_{机器}).$$

17.6　一　般　m

高维情形的分析是类似的. 例如，在5×5情形得到按分量的界

$$\frac{|\delta R|}{|R|} \leqslant \begin{bmatrix} 5 & 1 & 2 & 3 & 4 \\ & 4 & 1 & 2 & 3 \\ & & 3 & 1 & 2 \\ & & & 2 & 1 \\ & & & & 1 \end{bmatrix} \epsilon_{\text{机器}} + O(\epsilon_{\text{机器}}^2). \tag{17.14}$$

公式中矩阵的元素由 3 部分组成，乘法 $\tilde{x}_k r_{jk}$ 以模式

$$\otimes: \quad \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ & 0 & 1 & 1 & 1 \\ & & 0 & 1 & 1 \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix}. \tag{17.15}$$

引入扰动 $\epsilon_{\text{机器}}$. 除以 r_{kk} 以模式

$$\oslash: \quad \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}. \tag{17.16}$$

引入扰动. 最后，减法也出现在模式（17.15）中，而且由于决定由左到右计算，在对角线和在右边每个位置都引入一个扰动. 这些加起来直到模式

$$\ominus: \quad \begin{bmatrix} 4 & 0 & 1 & 2 & 3 \\ & 3 & 0 & 1 & 2 \\ & & 2 & 0 & 1 \\ & & & 1 & 0 \\ & & & & 0 \end{bmatrix}. \tag{17.17}$$

将（17.15），（17.16）和（17.17）加起来产生（17.14）的结果. 这就完成了定理 17.1 的证明.

126

17.7 注　　记

导出（17.14）的分析是所有浮点计算向后误差分析的典型. 惟一的原始要素是浮点公理（13.7），保证每个运算 $\oplus$，$\ominus$，$\otimes$ 或 $\oslash$（有时也包括浮点 $\sqrt{\ }$）引入小的相对误差. 反复且小心地应用这个公理，将计算过程中的每一个误差解释为初始数

据的一个误差. 阶$\epsilon_{机器}$的扰动附加地组合和在分子和分母之间自由地移动，因而差是$\epsilon_{机器}^2$阶的.

对一个给定的算法可以导出不只一个误差界. 在现在的情形下，我们可以和 r_{ij} 一样扰动 b_j，避免了对（17.17）表示技巧的需要. 另一方面，只有 $\boldsymbol{R}$ 被扰动这个最后的结论是十分清楚的.

方程（17.5）是一个按分量的（componentwise）向后误差界，意思是每个元素 r_{ij}被一个相对于它自己较小的量所扰动，它不是只相对于 $\boldsymbol{R}$ 的范数的. 例如，若 $r_{ij}=0$，则这个元素根本没有经历扰动：$\delta\boldsymbol{R}$ 有和 $\boldsymbol{R}$ 相同的稀疏模式. 某些数值线性代数的算法满足按分量向后误差估计，而某些则不是. 在后一种情形中我们一定要安排一个较弱的，如（17.4）那样的按范数（normwise）估计. 在第二次世界大战后数值分析早期的10年里，大多数的误差估计是由按范数的形式取得的，而在更近的这些年里，已经转向了按分量的分析，因为这样的结果是比较清晰的，而且满足按分量界的算法对变量的大小是更不灵敏的. 然而，本书中不讨论这些问题.

我们以一个评述来结束本节，这是关于像（17.5）或（17.14）那样的数量界和那些像（17.4）那样以 $O(\epsilon_{机器})$记号表示的界之间的关系的评述（习题 17.1）. 为什么不省去像（17.4）那样原始的描述? 理由是数量界一定包括$\sqrt{m}$或 m 那样的因子，这些是依顿范数的，不显著的，且由于统计相消在实际中常常是悲观的. 我们最好用$O(\epsilon_{机器})$表示我们大多数的结论以避免那样的复杂情况——最肯定地说，这是数值分析记住它们的形式.

习　　题

17.1　对范数$\|\cdot\|$任一特定的选择，界（17.5）隐含了一个比（17.4）更数量化的按范数的界. 对下列范数推导这样的界：

(a) $\|\cdot\|_1$，　(b) $\|\cdot\|_2$，　(c) $\|\cdot\|_\infty$.

127

17.2　用回代解三角形方程组（17.1），准确地说，定理 17.1 关于误差$\|\tilde{x}-x\|$隐含了什么?

17.3　令 $\boldsymbol{L}\in\mathbb{C}^{m\times m}$为一个单位下三角形矩阵（即对角线元素为 1 的下三角形矩阵）. 习惯上将 $\boldsymbol{L}$ 写为

$$\boldsymbol{L}=\begin{bmatrix} 1 & & & & \\ -\ell_{2,1} & 1 & & & \\ -\ell_{3,1} & -\ell_{3,2} & 1 & & \\ \vdots & \vdots & & \ddots & \\ -\ell_{m,1} & -\ell_{m,2} & -\ell_{m,3} & \cdots & 1 \end{bmatrix},$$

并定义 $\boldsymbol{M}=\boldsymbol{L}^{-1}$.

(a) 推导 m_{ij}的一个公式（它可以包含 $\boldsymbol{M}$ 的其他元素）. m_{ij}依赖于 $\boldsymbol{L}$ 的哪些元素?

(b) 假设 $\boldsymbol{L}$ 的次对角元素以相同的概率为独立的随机数 ±1. 固定 k，定义 $\mu_1=m_{kk}$，

$\mu_2 = m_{k+1,k}, \mu_3 = m_{k+2,k}, \cdots$，对数 μ_j 写出带有随机系数的递推关系组.

(c) 实验表明，元素 ± 1 的随机三角形矩阵是指数病态的，其意义是，若 κ_m 记为这类维数为 m 的矩阵的 2-范数条件数，则 $\lim_{m\to\infty}(\kappa_m)^{1/m} = C$ 对某常数 $1 < C < 1.5$ 成立.（极限过程可以用各种方法精确地进行，但我们不涉及这样的技术，认为它保持“概率 1”）. 完成包括各种维数随机矩阵的数值实验，来估计 C 达到 10% 或更好的准确度.

(d) 如果（c）的随机矩阵以（b）的随机序列 $\mu_1, \mu_2, \mu_3, \cdots$ 替代，更大量的实验是可行的. 解释（不证明）为什么常数 C 也可以由这些序列获得，进行数值实验来估计 C 达到 1% 或更好的准确度.

128

第 18 讲　最小二乘问题的条件

最小二乘问题的条件是一个微妙的课题，它将正方形方程组的条件与正交投影的几何性质结合起来. 因为它对最小二乘算法的稳定性有不平凡的内涵，所以它是非常重要的.

18.1　四个条件问题

在本讲我们转到线性最小二乘问题（11.2），并再次在图 18-1 中加以解释. 假设确定这个问题的矩阵是满秩的，且贯穿本讲的范数约为$\|\cdot\| = \|\cdot\|_2$：

$$\begin{aligned}&\text{给定满秩的 } \boldsymbol{A} \in \mathbb{C}^{m\times n}, m \geq n, \boldsymbol{b} \in \mathbb{C}^{m},\\ &\text{求 } \boldsymbol{x} \in \mathbb{C}^{n} \text{ 使得 } \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\| \text{最小.}\end{aligned} \tag{18.1}$$

解 $\boldsymbol{x}$ 及在 range$(\boldsymbol{A})$ 中最接近 $\boldsymbol{b}$ 的相应点由

$$\boldsymbol{x} = \boldsymbol{A}^{+}\boldsymbol{b}, \boldsymbol{y} = \boldsymbol{P}\boldsymbol{b}, \tag{18.2}$$

给定，其中 $\boldsymbol{A}^{+} \in \mathbb{C}^{n\times m}$是 $\boldsymbol{A}$ 的伪逆（11.11），$\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^{+} \in \mathbb{C}^{m\times m}$是到 range$(\boldsymbol{A})$的正交投影算子. 我们考虑（18.1）对于扰动的条件. 正如上一讲中显示了本书最详细的稳定性分析那样，现在这一讲显示我们最详细的条件数分析. 我们挑选最小二乘问题是因为其处理是有趣的，又因为它们有在下一讲中讨论的一个实际推论：作为一般最小二乘法的法方程组的不稳定性.

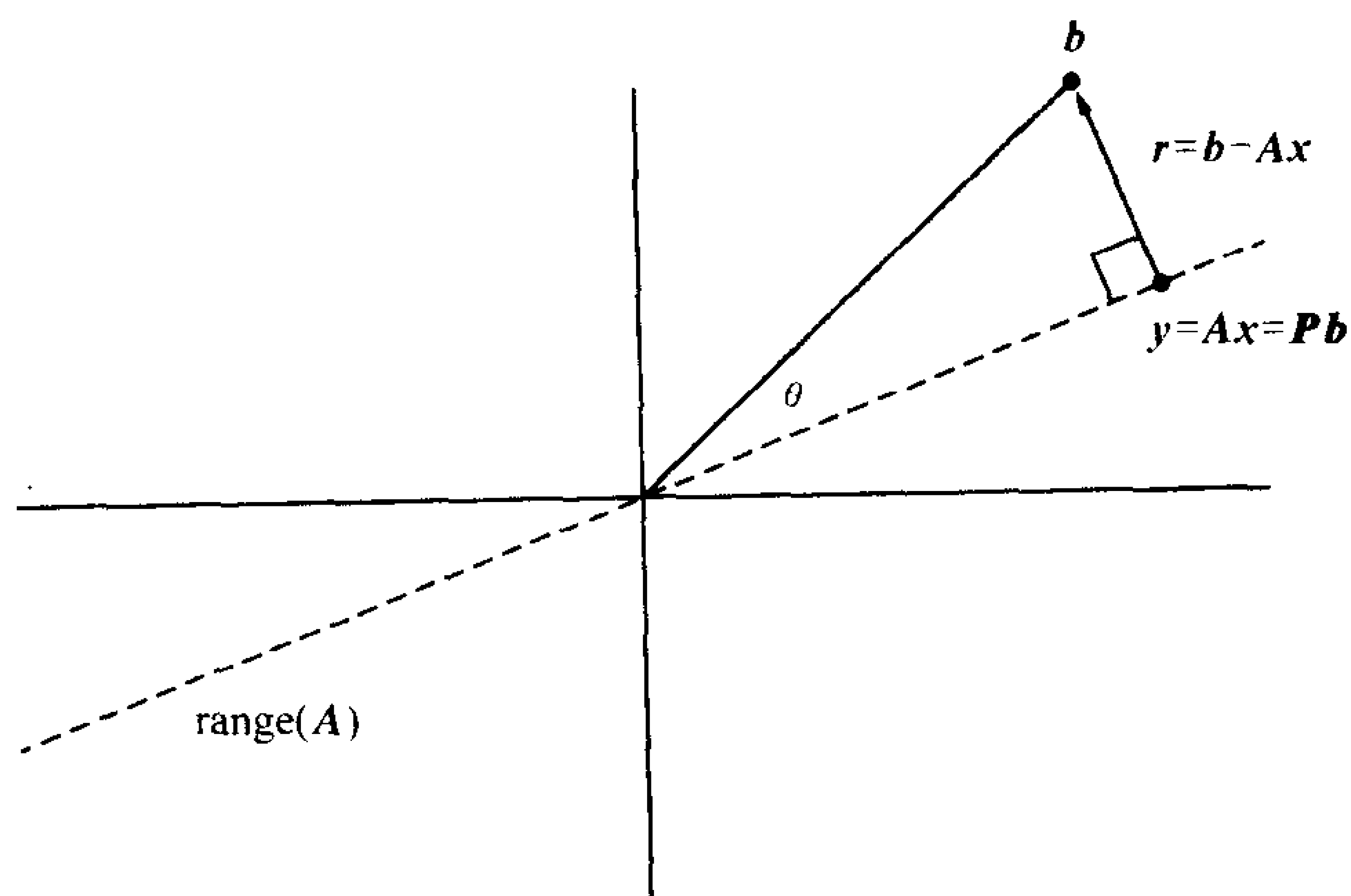


图 18-1　最小二乘问题（图 11-3 的重复）

条件与解对数据扰动的灵敏性有关. 对问题（18.1），我们将研究每个问题的两种选择. 问题的数据是 $m \times n$ 矩阵 $\boldsymbol{A}$ 和 m 维向量 $\boldsymbol{b}$. 解或者是系数向量 $\boldsymbol{x}$ 或者是对应的点 $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}$. 因此

$$\text{数据：}\boldsymbol{A},\ \boldsymbol{b},\qquad \text{解：}\boldsymbol{x},\ \boldsymbol{y}.$$

综合起来，这两对选择确定了我们要考虑的 4 个条件问题，所有这些问题都在某些范围中得到应用.

18.2　定　　理

本讲最重要的内容是定理 18.1，它提供了这些问题的答案. 结论用在最小二乘问题分析中重复出现的 3 个无量纲参数表示. 第 1 个是 $\boldsymbol{A}$ 的条件数. 对于正方形矩阵，这就是 $\kappa(\boldsymbol{A})=\|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|$，在矩形情形，定义由（12.17）产生，即

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}\|\|\boldsymbol{A}^{+}\| = \frac{\sigma_1}{\sigma_n}. \tag{18.3}$$

第 2 个是图 18-1 所画出的角 θ，它度量拟合的接近程度：

$$\theta = \cos^{-1}\frac{\|\boldsymbol{y}\|}{\|\boldsymbol{b}\|}. \tag{18.4}$$

130

第 3 个度量 $\|\boldsymbol{y}\|$ 是如何达不到它的最大可能值的量，给出 $\|\boldsymbol{A}\|$ 和 $\|\boldsymbol{x}\|$：

$$\eta = \frac{\|\boldsymbol{A}\|\|\boldsymbol{x}\|}{\|\boldsymbol{y}\|} = \frac{\|\boldsymbol{A}\|\|\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|}. \tag{18.5}$$

参数的范围是

$$1 \leqslant \kappa(\boldsymbol{A}) < \infty, \qquad 0 \leqslant \theta \leqslant \pi/2, \qquad 1 \leqslant \eta \leqslant \kappa(\boldsymbol{A}). \tag{18.6}$$

定理 18.1　固定 $\boldsymbol{b}\in\mathbb{C}^m$ 和满秩的 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$. 最小二乘问题（18.1）有下列 2-范数的相对条件数（12.5），描述 $\boldsymbol{y}$ 和 $\boldsymbol{x}$ 对 $\boldsymbol{b}$ 和 $\boldsymbol{A}$ 扰动的灵敏性：

	$\boldsymbol{y}$	$\boldsymbol{x}$
$\boldsymbol{b}$	$\dfrac{1}{\cos\theta}$	$\dfrac{\kappa(\boldsymbol{A})}{\eta\cos\theta}$
$\boldsymbol{A}$	$\dfrac{\kappa(\boldsymbol{A})}{\cos\theta}$	$\kappa(\boldsymbol{A})+\dfrac{\kappa(\boldsymbol{A})^2\tan\theta}{\eta}$

第一行的结论是准确的，在某个扰动 $\delta\boldsymbol{b}$ 达到，第二行的结论是上界.

在 $m=n$ 的特殊情形，（18.1）约化成一个正方形的非奇异方程组，有 $\theta=0$. 在此情形下，定理中第二列的数化成 $\kappa(\boldsymbol{A})/\eta$ 和 $\kappa(\boldsymbol{A})$，这是早先推导的结果（12.14）

和（12.18），而在左下方位置的数可以代之以 0（习题 18.4）.

18.3　变换到对角矩阵

作为定理 18.1 证明的第一步，注意到如果变换到一个方便的基，则最小二乘问题变得更易于分析. 令 $\boldsymbol{A}$ 有 $\boldsymbol{A}=\boldsymbol{U}\Sigma\boldsymbol{V}^*$ 形式的 SVD，其中 Σ 是一个 $m\times n$ 对角矩阵，它有正的对角元素. 因为扰动用 2-范数度量，它们的大小在基的酉变换下不变，所以 $\boldsymbol{A}$ 的扰动行为和 Σ 相同. 因此，不失一般性，我们可以直接对 Σ 进行处理. 在余下来的讨论中，我们假设 $\boldsymbol{A}=\Sigma$，并写成

$$\boldsymbol{A}=\begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \\ & & & \end{bmatrix}=\begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}. \tag{18.7}$$

131

这里 $\boldsymbol{A}_1$ 是一个 $n\times n$ 对角矩阵，$\boldsymbol{A}$ 的剩余部分为零.

$\boldsymbol{b}$ 在 range($\boldsymbol{A}$)上的正交投影现在是平凡的，写成

$$\boldsymbol{b}=\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{bmatrix},$$

其中 $\boldsymbol{b}_1$ 包含 $\boldsymbol{b}$ 的前 n 个元素. 所以投影 $\boldsymbol{y}=\boldsymbol{P}\boldsymbol{b}$ 为

$$\boldsymbol{y}=\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{0}\end{bmatrix}.$$

为了求对应的 $\boldsymbol{x}$，我们可写成 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}$，如

$$\begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{x}=\begin{bmatrix}\boldsymbol{b}_1\\ \boldsymbol{0}\end{bmatrix},$$

这隐含了

$$\boldsymbol{x}=\boldsymbol{A}_1^{-1}\boldsymbol{b}_1. \tag{18.8}$$

由这些公式，显然正交投影和伪逆是块 2×2 和 1×2 矩阵（参看习题 11.1）

$$\boldsymbol{P}=\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix},\qquad \boldsymbol{A}^+=\begin{bmatrix}\boldsymbol{A}_1^{-1} & \boldsymbol{0}\end{bmatrix}. \tag{18.9}$$

18.4 y 对 b 扰动的灵敏性

我们从 4 个条件结论中最简单的一个开始. 由（18.2），$\boldsymbol{b}$ 和 $\boldsymbol{y}$ 的关系正是线性方程组 $\boldsymbol{y}=\boldsymbol{P}\boldsymbol{b}$. 这个映射 $\boldsymbol{P}$ 的雅可比矩阵是 $\boldsymbol{P}$ 本身，且由（18.9）有 $\|\boldsymbol{P}\|=1$. 由（12.6）和（18.4），$\boldsymbol{y}$ 对扰动 $\boldsymbol{b}$ 的条件数因而是

$$\kappa_{b\mapsto y}=\frac{\|\boldsymbol{P}\|}{\|\boldsymbol{y}\|/\|\boldsymbol{b}\|}=\frac{1}{\cos\theta}.$$

这就形成了定理 18.1 左上方的结论. 对于满足 $\|\boldsymbol{P}(\delta\boldsymbol{b})\|=\|\delta\boldsymbol{b}\|$ 的扰动 $\delta\boldsymbol{b}$，条件数得到了体现（即达到了（12.5）的上确界），当 $\delta\boldsymbol{b}$ 除了前 n 个元素外为零的时候，这个情况出现了.

18.5 x 对 b 扰动的灵敏性

$\boldsymbol{b}$ 和 $\boldsymbol{x}$ 之间的关系也是线性的，$\boldsymbol{x}=\boldsymbol{A}^{+}\boldsymbol{b}$，其雅可比矩阵为 $\boldsymbol{A}^{+}$. 由式（12.6），式（18.4）和式（18.5），$\boldsymbol{x}$ 对 $\boldsymbol{b}$ 的扰动的条件数是

$$\kappa_{b\mapsto x}=\frac{\|\boldsymbol{A}^{+}\|}{\|\boldsymbol{x}\|/\|\boldsymbol{b}\|}=\|\boldsymbol{A}^{+}\|\frac{\|\boldsymbol{b}\|}{\|\boldsymbol{y}\|}\frac{\|\boldsymbol{y}\|}{\|\boldsymbol{x}\|}=\|\boldsymbol{A}^{+}\|\frac{1}{\cos\theta}\frac{\|\boldsymbol{A}\|}{\eta}=\frac{\kappa(\boldsymbol{A})}{\eta\cos\theta}.$$

这就形成了定理 18.1 右上方的结论. 这里，条件数由满足 $\|\boldsymbol{A}^{+}(\delta\boldsymbol{b})\|=\|\boldsymbol{A}^{+}\|\,\|\delta\boldsymbol{b}\|=\|\delta\boldsymbol{b}\|/\sigma_n$ 的扰动 $\delta\boldsymbol{b}$ 体现，这情况出现在 $\delta\boldsymbol{b}$ 除了第 n 个元素外为零的情况（或者也可能除了其他元素，这在 $\boldsymbol{A}$ 有多于一个奇异值等于 σ_n 的情况出现）. 132

18.6 range(A)的倾斜

$\boldsymbol{A}$ 中扰动的分析是一个非线性问题，而且更麻烦，可以由用代数方法计算雅可比矩阵开始，但代之的是，我们将用几何的观点. 出发点是观察 $\boldsymbol{A}$ 的扰动从两方面影响最小二乘问题：它们使 $\mathbb{C}^n$ 到 range($\boldsymbol{A}$)的映射失真，以及它们改变了 range($\boldsymbol{A}$)本身. 目前我们考虑后者的效果.

我们可以观察 range($\boldsymbol{A}$)空间小“倾斜”的轻微改变. 问题是，由 $\delta\boldsymbol{A}$ 的小扰动能给予的最大的倾斜角 $\delta\alpha$ 是什么？答案可如下确定. 在 $\boldsymbol{A}$ 映射下单位球面的像是一个平卧于 range($\boldsymbol{A}$)的超椭圆. 为了尽可能有效地改变 range($\boldsymbol{A}$)，我们抓住超椭圆上的一点 $\boldsymbol{p}=\boldsymbol{A}\boldsymbol{v}$（因此 $\|\boldsymbol{v}\|=1$）且在一个正交于 range($\boldsymbol{A}$)的方向 $\delta\boldsymbol{p}$ 上接近它. 最有效地达到这个目的的矩阵扰动是 $\delta\boldsymbol{A}=(\delta\boldsymbol{p})\boldsymbol{v}^{*}$，它给出（$\delta\boldsymbol{A}$）$\boldsymbol{v}=\delta\boldsymbol{p}$ 且 $\|\delta\boldsymbol{A}\|=\|\delta\boldsymbol{p}\|$（例 3.6）. 可见为了得到有给定 $\|\delta\boldsymbol{p}\|$ 的最大倾斜，应取 $\boldsymbol{p}$ 尽可能接近原点. 也就是要 $\boldsymbol{p}=\sigma_n\boldsymbol{u}_n$，其中 σ_n 是 $\boldsymbol{A}$ 的最小奇异值而 $\boldsymbol{u}_n$ 是对应的左奇异向量. 因有（18.7）的对角

形式，p 等于 A 的最后一列，v^* 是 n 维向量 $(0,0,\cdots,0,1)$，且 δA 是 A 在此列对角线以下元素的一个扰动. 这样的扰动以由 $\tan(\delta\alpha)=\|\delta p\|/\sigma_n$ 给出的 $\delta\alpha$ 倾斜 range(A). 因为 $\|\delta p\|=\|\delta A\|$ 及 $\delta\alpha\leqslant\tan(\delta\alpha)$，所以有

$$\delta\alpha\leqslant\frac{\|\delta A\|}{\sigma_n}=\frac{\|\delta A\|}{\|A\|}\kappa(A),\tag{18.10}$$

对于刚才所描述的选择 δA 等式成立，条件是它们是无穷小的 [这样 $\delta\alpha=\tan(\delta\alpha)$].

18.7　y 对 A 扰动的灵敏性

现在我们准备导出定理 18.1 表中的第二行，从它左边的元素开始. 因为 y 是 b 在 range(A)上的正交投影，所以它由 b 和 range(A)确定. 因而为分析 y 对 A 扰动的灵敏性，我们可以简单地研究以某个角 $\delta\alpha$ 倾斜的 range(A)的 y 上的效果.

当我们想像固定 b，观察 y 随着 range(A)的倾斜变化时（图 18-2），一个优美的几何性质就呈现在眼前了. 不管 range(A)怎样倾斜，向量 $y\in$range(A)总是正交于 $y-b$. 也就是直线 b-y 一定与直线 0-y 成直角. 换句话说，当 range(A)被调整时，y 沿着半径为 $\|b\|/2$，中点在 $b/2$ 的球面移动.

133

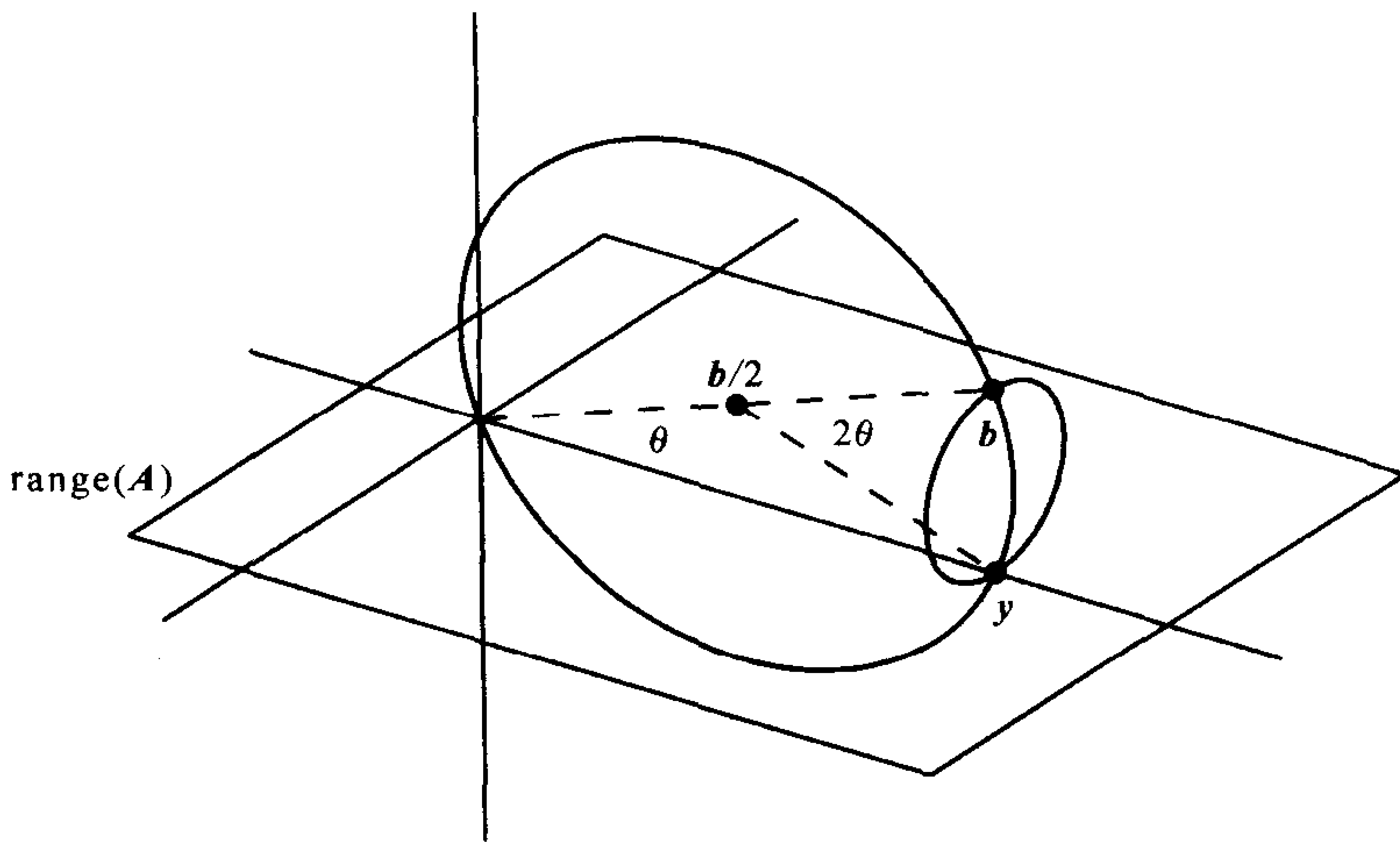


图 18-2　当 range（A）变化时，球面上 y 在其上移动的两个圆. 半径为 $\|b\|/2$ 的大圆对应在平面 0-b-y 倾斜 range(A)，半径为$(\|b\|/2)\sin\theta$ 的小圆对应在一个正交方向上倾斜它. 不管 range(A)如何倾斜，y 始终保持在半径为 $\|b\|/2$，中心在 $b/2$ 的球面上

在平面 0-b-y 上以角 $\delta\alpha$ 倾斜 range(A)，这在中心点 $b/2$ 以 $2\delta\alpha$ 改变角 2θ. 因而对应的扰动 δy 是中心角为 $2\delta\alpha$，边长为 $\|b\|/2$ 的等腰三角形的底. 这隐含了 $\|\delta y\|=\|b\|\sin(\delta\alpha)$. 以任何其他方向倾斜 range($A$)得到不同的平面中的相似几何性质和比

$\sin\theta$ 那样小的因子更小的扰动. 因而对以角 $\delta\alpha$ 任意的扰动有

$$\|\delta y\| \leqslant \|b\| \sin(\delta\alpha) \leqslant \|b\|\delta\alpha. \tag{18.11}$$

由式（18.4）和式（18.10），这给出 $\|\delta y\| \leqslant \|\delta A\|\kappa(A)\|y\|/\|A\|\cos\theta$, 即

$$\frac{\|\delta y\|}{\|y\|} \Big/ \frac{\|\delta A\|}{\|A\|} \leqslant \frac{\kappa(A)}{\cos\theta}. \tag{18.12}$$

这形成了定理 18.1 左下方的结论.

18.8　x 对 A 扰动的灵敏性

现在我们准备分析定理 18.1 中最令人感兴趣的关系：x 对 A 扰动的灵敏性.

很自然，一个扰动 δA 分为两部分：在 A 前 n 行的一部分 δA_1，及余下 $m-n$ 行的另一部分 δA_2：

$$\delta A = \begin{bmatrix} \delta A_1 \\ \delta A_2 \end{bmatrix} = \begin{bmatrix} \delta A_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \delta A_2 \end{bmatrix}.$$

134

首先考虑 δA_1 的效果. 这样的扰动改变了 A 在它值域的映射，但不改变 range(A) 本身或 y. 它在正方形方程组（18.8）以 δA_1 扰动 A_1，且没有改变 b_1. 对这样扰动的条件数由（12.18）给出，在这里写成

$$\frac{\|\delta x\|}{\|x\|} \Big/ \frac{\|\delta A_1\|}{\|A\|} \leqslant \kappa(A_1) = \kappa(A). \tag{18.13}$$

其次考虑（无穷小的）扰动 δA_2 的效果. 这样的扰动倾斜了 range(A) 而没有改变 A 在这个空间内的映射. 点 y 被扰动，因而向量 b_1 也被扰动，但 A_1 没有被扰动. 这对应于在（18.8）中扰动 b_1 而不改变 A_1. 对这样扰动的条件数由（12.14）给出，写成

$$\frac{\|\delta x\|}{\|x\|} \Big/ \frac{\|\delta b_1\|}{\|b_1\|} \leqslant \frac{\kappa(A_1)}{\eta(A_1;x)} = \frac{\kappa(A)}{\eta}. \tag{18.14}$$

为了完成论证，需要将 δb_1 与 δA_2 联系起来. 现在向量 b_1 是 y 在 range(A) 的坐标中的表示. 因而由在 b_1 中的变化实现的在 y 中的惟一变化是那些平行于 range(A) 的变化，正交的变化没有作用. 特别地，如果 range(A) 以角 $\delta\alpha$ 在平面 **0-*b*-*y*** 上倾斜，那么所得的扰动 δy 不平行于 range(A)，而是有一个角 $\pi/2-\theta$. 结果是 b_1 的改变满足 $\|\delta b_1\| = \sin\theta\|\delta y\|$. 由（18.11），有

$$\|\delta b_1\| \leqslant (\|b\|\delta\alpha)\sin\theta. \tag{18.15}$$

奇怪的是，如果 range(A) 以一个正交于平面 **0-*b*-*y*** 的方向倾斜，那么会以不同的

理由得到同样的界. 现在δy平行于range($\boldsymbol{A}$)，但它有比$\sin\theta$更小的因子，如上与图18-2联系的讨论. 因而$\|\delta \boldsymbol{y}\| \leqslant (\|\boldsymbol{b}\|\delta\alpha)\sin\theta$，且由$\|\delta \boldsymbol{b}_1\| \leqslant \|\delta \boldsymbol{y}\|$，再次得到(18.15).

现在万事俱备. 因$\|\boldsymbol{b}_1\| = \|\boldsymbol{b}\|\cos\theta$，所以将（18.15）重写为

$$\frac{\|\delta \boldsymbol{b}_1\|}{\|\boldsymbol{b}_1\|} \leqslant (\delta\alpha)\tan\theta. \tag{18.16}$$

由（18.10），将$\delta\alpha$与$\|\delta A_2\|$联系并组合（18.14）和（18.16），得到

$$\frac{\|\delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|}\bigg/\frac{\|\delta \boldsymbol{A}_2\|}{\|\boldsymbol{A}\|} \leqslant \frac{\kappa(\boldsymbol{A})^2\tan\theta}{\eta}.$$

135　将此结果加到（18.13）就形成了定理18.1右下方的结论.

习　　题

18.1　考虑例子

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \\ 1 & 1.0001 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 2 \\ 0.0001 \\ 4.0001 \end{bmatrix}.$$

(a) 本例的矩阵$\boldsymbol{A}^+$和$\boldsymbol{P}$是什么？给出准确的答案.

(b) 求最小二乘问题$\boldsymbol{Ax} \approx \boldsymbol{b}$的准确答案$\boldsymbol{x}$和$\boldsymbol{y} = \boldsymbol{Ax}$.

(c) $\kappa(\boldsymbol{A})$，θ和η是什么？从这开始，数值答案是可以接受的.

(d) 定理18.1的4个条件数是什么？

(e) 给出近似达到这4个条件数的扰动$\delta\boldsymbol{b}$和$\delta\boldsymbol{A}$的例子.

18.2　研究社会学的人信赖回归(regression)技术，在其中，由某些量的观测值组成的一个向量在最小二乘意义下用另外向量的线性组合来近似. 比如，拟合的系数被解释为某些因素象征，如IQ、教育的年限、双亲的教育年限和双亲的收入对年收益的影响.

可以想像包含在这样的模型中的变元越多，就会获得越多的信息，但这并非总是正确的，用条件的观点解释这种现象，特别参照定理18.1的结论.

18.3　假设你在湖边观看对岸一所房子发出的光. 如果湖面泛起细浪，反射光作为垂直的条纹出现. 同样的效果出现在雨天路上你前面汽车的尾灯上，或者甚至是过道的灯光在上了蜡的光亮地板上的反射. 这是一个实际的效果，不是一个光学解释，它是几何的内容.

(a) 导出一个数量的理论解释这个现象. 特别地，假设你和湖对岸的房子都在湖面50米之上，湖面有一千米宽. 出现在你视觉范围内的条纹的长度对宽度的比是多少？

(b) 描述这个问题和本讲的一个几何讨论之间的联系.

136　18.4　解释为什么$\boldsymbol{y}$对$\boldsymbol{A}$扰动的条件数在$m = n$的情形成为0，把它作为定理18.1的注记.

第 19 讲　最小二乘算法的稳定性

最小二乘问题可以用第 11 讲中描述的各种方法求解，包括法方程组、豪斯霍尔德三角形化、格拉姆-施密特正交化和 SVD. 这里我们比较这些方法并说明法方程组的使用一般是不稳定的.

19.1　例　　子

为了说明算法的性质，我们将它们应用到一个 $m=100$，$n=15$ 的数值例子，这里是 MATLAB 的程序：

```
m = 100;n = 15;
t = (0:m-1)'/(m-1);                  令 t 为[0,1]的一个离散化.
A = [];for i = 1:n,                  构造范德蒙德矩阵.
  A = [A t.^(i-1)]; end
b = exp(sin(4*t));                   右边.
b = b/2006.787453080206;             正规化(参见正文).
```

这个例子背后的思想是在区间 [0,1] 上用一个 14 次多项式最小二乘拟合函数 exp (sin (4τ)). 首先离散 [0,1]，定义由 0 到 1 的 100 个等距点的向量 $\boldsymbol{t}$. 矩阵 $\boldsymbol{A}$ 是 100×15 范德蒙德矩阵，其列是幂 $1,\tau,\cdots,\tau^{14}$ 在 $\boldsymbol{t}$ 的点上的样本，右边的 $\boldsymbol{b}$ 是函数 exp (sin (4τ)) 在这些点上的样本.

程序编码最后一行古怪的写法基于如下理由. 为简单起见，我们只比较由各种算法计算出的系数 x_{15}. 没有这最后一行，x_{15} 的准确值会是 2006.787-453-080-206…(这些数字是由一个扩展精度的算法程序包得到的). 除以这个数，我们得到一个其解有 $x_{15}=1$ 的问题，这使我们的比较更容易进行.

为了解释我们的观察，我们需要量 (18.3) ~ (18.5). 可以通过充分准确地数值解最小二乘问题并借用 MATLAB 的 \ 运算这一工具来确定这些量：

```
x = A\b;y = A*x;                          解最小二乘问题
kappa = cond(A)
   kappa = 2.2718e+10                     κ(A)
theta = asin(norm(b-y)/norm(b))
   theta = 3.7461e-06                     θ
eta = norm(A)*norm(x)/norm(y)
   eta = 2.1036e+05                       η
```

结论 $\kappa(A)\approx 10^{10}$ 指出单项式 $1,t,\cdots,t^{14}$ 组成高度病态的基. 结论 $\theta\approx 10^{-6}$ 指出 $\exp(\sin(4t))$ 可以用一个 14 次多项式非常接近地拟合. (拟合是如此接近，以致于我们用公式 $\theta=\sin^{-1}(\|b-y\|/\|b\|)$ 代替 (18.4)，以避免相消误差.) 对于 η，它的值约为 10^5，大约位于 (18.6) 所允许的极端值 1 和 $\kappa(A)$ 的中间.

将这些数代入到定理 18.1 的公式中，对我们的例子问题，我们求得 $\boldsymbol{y}$ 和 $\boldsymbol{x}$ 对于 $\boldsymbol{b}$ 和 $\boldsymbol{A}$ 扰动的条件数近似为

	$\boldsymbol{y}$	$\boldsymbol{x}$
$\boldsymbol{b}$	1.0	1.1×10^5
$\boldsymbol{A}$	2.3×10^{10}	3.2×10^{10}

19.2 豪斯霍尔德三角形化

如第 11 讲指出，解最小二乘问题的标准算法是借助豪斯霍尔德三角形化的 QR 因子分解 (算法 11.2). 这里是一个 MATLAB 实验得到的结果：

```
[Q,R] = qr(A,0);          A 的豪斯霍尔德三角形化.
x = R\(Q'*b);             解 x.
x(15)
  ans = 1.00000031528723
```

138

怎样理解这个结果? 考虑到正规化，正确的答案会是 $x_{15}=1$. 因此我们有大约 3×10^7 的相对误差. 因为计算是在 $\epsilon_{机器}\approx 10^{-16}$ 的 IEEE 双精度运算下进行的，这意味着舍入误差已经被一个 10^9 阶的因子放大. 乍看起来这是坏事，但细看一下上面的表，它提醒我们 $\boldsymbol{x}$ 对 $\boldsymbol{A}$ 扰动的条件数是 10^{10} 阶. 因此 x_{15} 的不准确可以完全用病态来解释，而不是不稳定引起的. 算法 11.2 出现了向后稳定性.

上面显式地得到 $\hat{Q}$，但正如第 10 和第 16 讲所强调的，这并非是必要的. 存储算法 10.1 的第 k 步所决定的向量 v_k (方程 (10.5)) 就足够了，由算法 10.2，它们可以用来计算 $\hat{Q}^*b$. 在 MATLAB 中，我们可以由计算 QR 因子分解达到这个效果，这不仅仅是 $\boldsymbol{A}$ 的也可以是 $m\times(n+1)$ “增广矩阵” $[\boldsymbol{A}\ \boldsymbol{b}]$ 的 QR 因子分解. 在分解的过程中，使 $\boldsymbol{A}$ 成上三角形的豪斯霍尔德镜射算子也应用到 $\boldsymbol{b}$ 上，产生的向量 $\hat{Q}^*b$ 在第 $n+1$ 列的前 n 个位置. 一个附加的第 $n+1$ 个镜射算子用来使列 $n+1$ 的 $n+2,\cdots,m$ 的元素为零，但这并不改变这列的前 n 个元素，这是我们关注的元素. 因而

`[Q,R] = qr([A b],0);`	$[A\ b]$的豪斯霍尔德三角形化
`Qb = R(1:n,n+1);`	提取$\hat{Q}^*b$ ...
`R = R(1:n,1:n);`	...和$\hat{R}$.
`x = R\Qb;`	解 x.
`x(15)`	
`   ans = 1.00000031529465`	

答案几乎和前面一样. 这指出在 A 的 QR 因子分解中引入的误差淹没了$\hat{Q}^*b$ 计算所引入的误差.

在 MATLAB 中，也存在借助豪斯霍尔德三角形化解最小二乘问题的第 3 种方法. 我们可以用内置运算 \ ，如同在求 $\kappa(A)$，θ 和 η 中已经做过的那样：

`x = A\b;`	解 x.
`x(15)`	
`   ans = 0.99999994311087`	

这个结果显然不同于其他结果，准确度多了一个数量阶. 理由是 MATLAB 的 \ 运算用了列选主元的 QR 因子分解（QR factorization with column pivoting），它基于因子分解 $AP=\hat{Q}\hat{R}$，其中 P 是一个置换矩阵. 本书中将不讨论列选主元的问题.

139

由按范数稳定性分析的观点，这三个 QR 因子分解的变种是等同的. 可以证明所有这三者都是向后稳定的.

定理 19.1　让满秩最小二乘问题（11.2）在一个满足（13.5）和（13.7）的计算机上用豪斯霍尔德三角形化（算法 11.2）求解. 这个算法在计算的解 $\tilde{x}$ 对某个 $\delta A\in\mathbb{C}^{m\times n}$ 具有性质

$$\|(A+\delta A)\tilde{x}-b\| = \min, \qquad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{机器}}) \tag{19.1}$$

的意义下是向后稳定的. 不论$\hat{Q}^*b$ 是借助$\hat{Q}$的显式公式计算还是由算法 10.2 隐式计算都是正确的. 它也对任意列选主元的豪斯霍尔德三角形化成立.

19.3　格拉姆-施密特正交化

解最小二乘问题的另一种方法是修正的格拉姆-施密特正交化（算法 8.1）. 对 $m\approx n$，这比豪斯霍尔德正交化要多一点运算，但对 $m\gg n$，两个算法的浮点运算计数都渐近于 $2mn^2$.

下面的 MATLAB 序列以显然的方式实现这个算法. 函数 mgs 是算法 8.1 的一个实现（不显示）——与第 9 讲的实验 2 相同.

```
[Q,R] = mgs(A);            A 的格拉姆-施密特正交化.
x = R\(Q'*b);              解 x.
x(15)
  ans = 1.02926594532672
```

结果十分差. 舍入误差以 10^{14} 阶的因子放大，远大于问题的条件数. 事实上，此算法是不稳定的，其理由容易解释. 如同在第 9 讲末尾所指出的一样，格拉姆-施密特正交化一般产生矩阵$\hat{Q}$，它的列不是精确正交的. 因为上述算法依赖于那里的正交性，所以它也遭遇相应的情况.

将算法的公式重新安排可以避免不稳定性. 因为格拉姆-施密特迭代提供了一个准确的乘积$\hat{Q}\hat{R}$，即使$\hat{Q}$并没有准确的正交列，一个途径是对向量 Rx 建立法方程组 $Rx=(\hat{Q}^*\hat{Q})^{-1}\hat{Q}^*b$，再由回代求得 x. 只要计算的$\hat{Q}$至少是良态的，这个方法将摆脱上述法方程组用于任意矩阵的不稳定性. 然而，它包含了不必要的额外的工作量，所以在实际中不应该采用.

140

一个使格拉姆-施密特稳定化更好的方法是使用增广方程组，就像上述两个豪斯霍尔德实验的第 2 个实验那样.

```
[Q,R] = mgs([A b]);        [A b]的格拉姆-施密特正交化.
Qb = R(1:n,n+1);           提取Q̂*b ...
R = R(1:n,1:n);                ...和R̂.
x = R\Qb;                  解 x.
x(15)
  ans = 1.00000005653399
```

现在结果看起来和豪斯霍尔德三角形化一样好. 可以证明情况总是这样的.

定理 19.2 若$\hat{Q}^*b$ 如同在上面程序段所指出的那样，隐式构成利用格拉姆-施密特正交化求出的满秩最小二乘问题 (11.2) 的解也是向后稳定的，满足 (19.1).

19.4 法方程组

与最小二乘问题本质上不同的另一个方法是解法方程组（算法 11.1），典型的解法是楚列斯基因子分解（第 23 讲）. 对 $m\gg n$，这个方法比依赖于显式正交化的方法快 2 倍，渐近地需要 mn^2 次 flop (11.14). 在下面的实验中，这个问题用 \ 运算的单

行 MATLAB 求解：

```
x = (A' * A) \ (A' * b);                组成和求解法方程组.
x(15)
  ans = 0.39339069870283
```

结果十分糟糕！它是我们得到的最坏结果，甚至没有一个数字位的准确度，法方程组的使用对解最小二乘问题显然是不稳定的方法. 我们将花一点时间解释这个现象，因为这个解释是条件与稳定性思想相互作用的一个完整的例子. 同时，法方程组是如此经常被人使用，以致于了解它所包含的危险是重要的.

假设对满秩问题（11.2）有一个向后稳定的算法，它提交了一个解 $\tilde{x}$，满足 $\|(A+\delta A)\tilde{x}-b\|=\min$ 对某个满足 $\|\delta A\|/\|A\|=O(\epsilon_{机器})$ 的 δA 成立.（允许 b 和 A 同样扰动，或考虑用稳定性代替向后稳定性，这都不改变我们的主要观点.）由定理 15.1 和 18.1 有

$$\frac{\|\tilde{x}-x\|}{\|x\|}=O\left(\left(\kappa+\frac{\kappa^2\tan\theta}{\eta}\right)\epsilon_{机器}\right), \tag{19.2}$$

其中 $\kappa=\kappa(A)$. 现在假设 A 病态，即 $\kappa\gg1$，且 θ 被界于远离 $\pi/2$. 依赖于各种参数值，会发生两种非常不同的情况. 若 $\tan\theta$ 为阶 1（即最小二乘拟合不是特别接近）且 $\eta\ll\kappa$，则（19.2）右边为 $O(\kappa^2\epsilon_{机器})$，另一方面，若 $\tan\theta$ 接近零（一个非常接近的拟合）或者 η 接近 κ，其界为 $O(\kappa\epsilon_{机器})$. *最小二乘问题的条件数可以位于 κ 到 κ^2 范围的任意处.*

现在考虑用法方程组 $(A^*A)x=A^*b$ 解（11.2）时会发生什么事情. 对这个方程组楚列斯基因子分解是一个稳定的算法，其意义是它产生的解 $\tilde{x}$，对某个满足 $\|\delta H\|/\|A^*A\|=O(\epsilon_{机器})$ 的 δH，满足 $(A^*A+\delta H)\tilde{x}=A^*b$（定理 23.3）. 然而，矩阵 A^*A 的条件数是 κ^2 而不是 κ. 因此由法方程组可期望的最好结果是

$$\frac{\|\tilde{x}-x\|}{\|x\|}=O(\kappa^2\epsilon_{机器}). \tag{19.3}$$

法方程组的性质由 κ^2 而不是由 κ 控制.

现在结论清楚了. 如果 $\tan\theta$ 是阶 1 的且 $\eta\ll\kappa$，或如果 κ 是阶 1，则（19.2）和（19.3）为同阶的且法方程组是稳定的，然而，如果 κ 大且不论 $\tan\theta$ 接近零还是 η 接近 κ，那么（19.3）比（19.2）大很多，且法方程组不稳定. *对于含接近拟合的病态问题，法方程组是典型不稳定的.* 在我们的例子中，$\kappa^2\approx10^{20}$，所以楚列斯基因子分解没有产生准确数字位并不会使人感到惊奇.

根据我们的定义，只有当一个算法对所有考虑的问题的有令人满意的一致性质

时，这个算法才是稳定的. 因而下面的结论是刚才所做观察的形式化的叙述.

定理 19.3 满秩最小二乘问题（11.2）用法方程组求得的解（算法 11.1）是不稳定的. 然而，通过在问题中限制 $\kappa(A)$ 为一致上有界或 $(\tan\theta)/\eta$ 为一致下有界，稳定性就可以达到.

19.5 SVD

对最小二乘问题，更进一步的算法是在第 11 讲中提到的用 SVD 的算法（算法 11.3）. 像大多数基于 SVD 的计算一样，这个算法是稳定的：

```
[U,S,V] = svd(A,0);          A 的约化 SVD.
x = V * (S\(U' * b));        解 x.
x(15)
   ans = 0.99999998230471
```

142

事实上，这是在我们的实验中得到的所有结果中最准确的一个，它以一个约为 3 的因子胜过列选主元的豪斯霍尔德三角形化（MATLAB 的 \ ）. 可以证明下面用通常形式表示的定理.

定理 19.4 用 SVD（算法 11.3）解满秩的最小二乘问题（11.2）所得的解是向后稳定的，满足估计（19.1）.

19.6 秩亏损最小二乘问题

在本讲中已经辨识了线性最小二乘问题的 4 种向后稳定的算法：豪斯霍尔德三角形化、列选主元的豪斯霍尔德三角形化、隐式计算 $\hat{Q}^* b$ 的修正格拉姆-施密特以及 SVD. 从对满秩问题（11.2）用经典的按范数稳定性分析的观点来看，这些算法中的差别是次要的，所以可以和用其他方法一样使用最简单、最经济的方法，即没有选主元的豪斯霍尔德三角形化.

然而，存在其他类型的最小二乘问题，在其中列选主元和 SVD 有特殊的重要性. 这些是 A 有 rank $< n$ 也可能是 $m < n$ 的问题，这样使得方程组是欠定的（underdetermined）. 这样的问题没有惟一解，除非加上附加的条件，典型的条件是 x 自身有尽可能小的范数. 更复杂的情况是，正确的解依赖于 A 的秩，而在舍入误差的情况下，确定秩的数值从来不是简单的事.

因此秩亏损的最小二乘问题不是最小二乘问题的一个具有挑战性的子类，而是基本不同的问题. 因为解的定义是新的，没有理由认为一个算法对满秩问题稳定就一定对秩亏损情形也稳定. 事实上，对秩亏损问题惟一完全稳定的算法是那些基于

SVD 的算法. 另外一个就是列选主元的豪斯霍尔德三角形化，它对几乎所有问题都稳定. 我们不给出详细的讨论.

习　题

19.1　给出秩为 n 的 $A \in \mathbb{C}^{m\times n}$ 及 $b \in \mathbb{C}^{m}$，考虑块 2×2 方程组

$$\begin{bmatrix} I & A \\ A^* & 0 \end{bmatrix}\begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{19.4}$$

其中 I 是 $m\times m$ 单位矩阵. 证明方程组有惟一解 $(r,x)^{\mathrm{T}}$，且向量 r 和 x 是剩余向量和最小二乘问题（18.1）的解. 143

19.2　这是一个 MATLAB 内置的 m 文件.

```
[U,S,V] = svd(A);
S = diag(S);
tol = max(size(A)) * S(1) * eps;
r = sum(S > tol);
S = diag(ones(r,1) ./S(1:r));
X = V(:,1:r) * S * U(:,1:r)';
```

144

这个程序计算什么?

第Ⅳ部分

方　程　组

第20讲　高斯消元法

对多数读者来说，高斯消元法无疑是熟悉的，它是手算解线性方程组最简单的方法，也是在计算机上解线性方程组的标准方法. 我们首先叙述高斯消元法的纯粹形式，然后在下一讲中，再加上选主元行部分——对稳定性至关重要的.

20.1　LU因子分解

高斯消元法通过在左边应用简单的线性变换，把一个满线性方程组变换为一个上三角形方程组. 这里的考虑类似于计算QR分解的豪斯霍尔德三角形化，不同之处是应用在高斯消元法的变换不是酉的.

令 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$ 为一个正方形矩阵.（这个算法也可用在矩形矩阵上，但是实际上很少有这样做的，我们将注意力限定在正方形情形.）我们的想法是，首先在第1列，然后第2列，如此继续下去，在对角线以下引入零元素——如同在豪斯霍尔德三角形化那样，从而把 $\boldsymbol{A}$ 变换成一个 $m\times m$ 的上三角形矩阵 $\boldsymbol{U}$. 这是由每一行后面的行减去该行的倍数来实现的. 这个"消元"的过程等价于用一系列的下三角形矩阵 $\boldsymbol{L}_k$ 乘在 $\boldsymbol{A}$ 的左边：

$$\underbrace{\boldsymbol{L}_{m-1}\cdots\boldsymbol{L}_2\boldsymbol{L}_1}_{\boldsymbol{L}^{-1}}\boldsymbol{A}=\boldsymbol{U}. \tag{20.1}$$

147

令 $\boldsymbol{L}=\boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{m-1}^{-1}$，得到 $\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}$. 这样我们得到 $\boldsymbol{A}$ 的*LU因子分解*（LU factorization）

$$\boldsymbol{A}=\boldsymbol{L}\boldsymbol{U}, \tag{20.2}$$

其中 $\boldsymbol{U}$ 是上三角形的，而 $\boldsymbol{L}$ 是下三角形的. 得到的结果是 $\boldsymbol{L}$ 为*单位下三角形的*（unit lower-triangular），意思是它的所有对角元素都等于1.

例如，设从一个 4×4 矩阵开始，算法分3步进行（比较（10.1））：

$$\underset{\boldsymbol{A}}{\begin{bmatrix}\times&\times&\times&\times\\\times&\times&\times&\times\\\times&\times&\times&\times\\\times&\times&\times&\times\end{bmatrix}}\xrightarrow{\boldsymbol{L}_1}\underset{\boldsymbol{L}_1\boldsymbol{A}}{\begin{bmatrix}\times&\times&\times&\times\\\mathbf{0}&\boldsymbol{\times}&\boldsymbol{\times}&\boldsymbol{\times}\\\mathbf{0}&\boldsymbol{\times}&\boldsymbol{\times}&\boldsymbol{\times}\\\mathbf{0}&\boldsymbol{\times}&\boldsymbol{\times}&\boldsymbol{\times}\end{bmatrix}}\xrightarrow{\boldsymbol{L}_2}\underset{\boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{A}}{\begin{bmatrix}\times&\times&\times&\times\\&\times&\times&\times\\&\mathbf{0}&\boldsymbol{\times}&\boldsymbol{\times}\\&\mathbf{0}&\boldsymbol{\times}&\boldsymbol{\times}\end{bmatrix}}\xrightarrow{\boldsymbol{L}_3}\underset{\boldsymbol{L}_3\boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{A}}{\begin{bmatrix}\times&\times&\times&\times\\&\times&\times&\times\\&&\times&\times\\&&\mathbf{0}&\boldsymbol{\times}\end{bmatrix}}.$$

（同在第10讲中一样，黑粗体表示刚刚运算过的元素，空白元素为零.）第 k 次变换 $\boldsymbol{L}_k$ 由第 $k+1,\cdots,m$ 行减去第 k 行的倍数，来引入第 k 列对角线以下的零. 因为第 k 行开头 $k-1$ 个元素已经为零，所以这个运算不破坏前面引入的任何零元素.

高斯消元法因而扩张了矩阵因子分解算法的类别：

格拉姆-施密特：$A = QR$　由三角形的正交化得到，
豪斯霍尔德：$A = QR$　由正交的三角形化得到，
高斯消元法：$A = LU$　由三角形的三角形化得到.

20.2 例　　子

为讨论细节，我们举一个示范的数值例子. 假设开始是一个 4×4 的矩阵

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}. \tag{20.3}$$

（A 的元素决不是随机的，它们被选来给出一个简单的 LU 因子分解.）高斯消元法的第一步：

148

$$L_1A = \begin{bmatrix} 1 & & & \\ -2 & 1 & & \\ -4 & & 1 & \\ -3 & & & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix}.$$

也就是说，我们已由第二行减去第一行的 2 倍，第三行减去 4 乘第一行，第四行减去 3 乘第一行. 第二步为：

$$L_2L_1A = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & -3 & 1 & \\ & -4 & & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & 2 & 4 \end{bmatrix}.$$

这次我们由第三行减去 3 乘第二行，由第四行减去 4 乘第二行. 最后，第三步我们由第四行减去第三行：

$$L_3L_2L_1A = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & & 2 \end{bmatrix} = U.$$

现在为了显示完全的因子分解 $A=LU$，需要计算乘积 $L=L_1^{-1}L_2^{-1}L_3^{-1}$. 也许会令人惊奇，结果是平凡的，$L_1$ 的逆正好是 L_1 本身，但是每个对角线以下的元素都取相反数：

$$\begin{bmatrix}1&&&\\-2&1&&\\-4&&1&\\-3&&&1\end{bmatrix}^{-1}=\begin{bmatrix}1&&&\\2&1&&\\4&&1&\\3&&&1\end{bmatrix}. \tag{20.4}$$

类似地，L_2 和 L_3 的逆由它们对角线以下元素取相反数得到. 最后，乘积 $L_1^{-1}L_2^{-1}L_3^{-1}$ 正好是具有 L_1^{-1}, L_2^{-1} 和 L_3^{-1} 的非零对角线以下元素嵌入到适当位置的单位下三角形矩阵. 总起来有

$$\underbrace{\begin{bmatrix}2&1&1&0\\4&3&3&1\\8&7&9&5\\6&7&9&8\end{bmatrix}}_{A}=\underbrace{\begin{bmatrix}1&&&\\2&1&&\\4&3&1&\\3&4&1&1\end{bmatrix}}_{L}\underbrace{\begin{bmatrix}2&1&1&0\\&1&1&1\\&&2&2\\&&&2\end{bmatrix}}_{U}. \tag{20.5}$$

20.3　一般形式和两个幸运之处

这里是 $m\times m$ 矩阵的一般形式，设 x_k 表示第 k 步开始时矩阵的第 k 列，则选择变换 L_k 要使得： 149

$$x_k=\begin{bmatrix}x_{1k}\\\vdots\\x_{kk}\\x_{k+1,k}\\\vdots\\x_{mk}\end{bmatrix}\xrightarrow{L_k}L_kx_k=\begin{bmatrix}x_{1k}\\\vdots\\x_{kk}\\0\\\vdots\\0\end{bmatrix}.$$

为了做到这点，希望从第 j 行减去 ℓ_{jk}乘第 k 行，其中 ℓ_{jk}是乘数(multiplier)

$$\ell_{jk}=\frac{x_{jk}}{x_{kk}}\quad(k<j\leqslant m). \tag{20.6}$$

矩阵 L_k 取为

$$L_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -\ell_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -\ell_{mk} & & & 1 \end{bmatrix},$$

它在第 k 列有非零的对角线以下元素，这类似于豪斯霍尔德三角形化的（10.2）.

在上面的数值例子中，我们注意有两个幸运之处：L_k 的逆可以用它对角线以下的元素取相反数得到（20.4），而且 L 可以由将元素 ℓ_{ik} 收集到适当的位置来组成（20.5）. 我们可以如下解释这些好运气的片段，定义

$$\ell_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \ell_{k+1,k} \\ \vdots \\ \ell_{m,k} \end{bmatrix}.$$

则 L_k 可以写成 $L_k = I - \ell_k e_k^*$，其中 e_k 像通常那样表示在位置 k 处为 1 而其他处为 0 的向量. ℓ_k 的稀疏形式隐含了 $e_k^* \ell_k = 0$，因而 $(I - \ell_k e_k^*)(I + \ell_k e_k^*) = I - \ell_k e_k^* \ell_k e_k^* = I$. 换句话说，$L_k$ 的逆是 $I + \ell_k e_k^*$，如在（20.4）中所示.

对第二个幸运之处说明如下，例如考虑乘积 $L_k^{-1} L_{k+1}^{-1}$，由 ℓ_{k+1} 的稀疏形式，有 $e_k^* \ell_{k+1} = 0$，因而

150

$$L_k^{-1} L_{k+1}^{-1} = (I + \ell_k e_k^*)(I + \ell_{k+1} e_{k+1}^*) = I + \ell_k e_k^* + \ell_{k+1} e_{k+1}^*.$$

因此 $L_k^{-1} L_{k+1}^{-1}$ 是单位下三角形矩阵，它正好是把 L_k^{-1} 和 L_{k+1}^{-1} 两者的元素嵌入到它们在对角线以下通常的位置. 当我们取所有这些矩阵的乘积组成 L 时，在对角线下面的每处都有同样方便的性质：

$$L = L_1^{-1} L_2^{-1} \cdots L_{m-1}^{-1} = \begin{bmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \ell_{31} & \ell_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{m1} & \ell_{m2} & \cdots & \ell_{m,m-1} & 1 \end{bmatrix}. \qquad (20.7)$$

尽管我们在第 8 讲并未指出，但是稀疏性的考虑导出了（20.7）的情况也出现在修正格拉姆-豪斯霍尔德过程的说明中（8.10），该过程是逐次右乘以三角形矩阵 R_k 来实现的.

在实际的高斯消元法中，矩阵 L_k 从来不是显式地形成和作乘法的. 乘数 ℓ_{jk} 被计

算且直接存到 $\boldsymbol{L}$ 中，因而变换 $\boldsymbol{L}_k$ 是隐式应用的.

算法 20.1　不选主元素的高斯消元

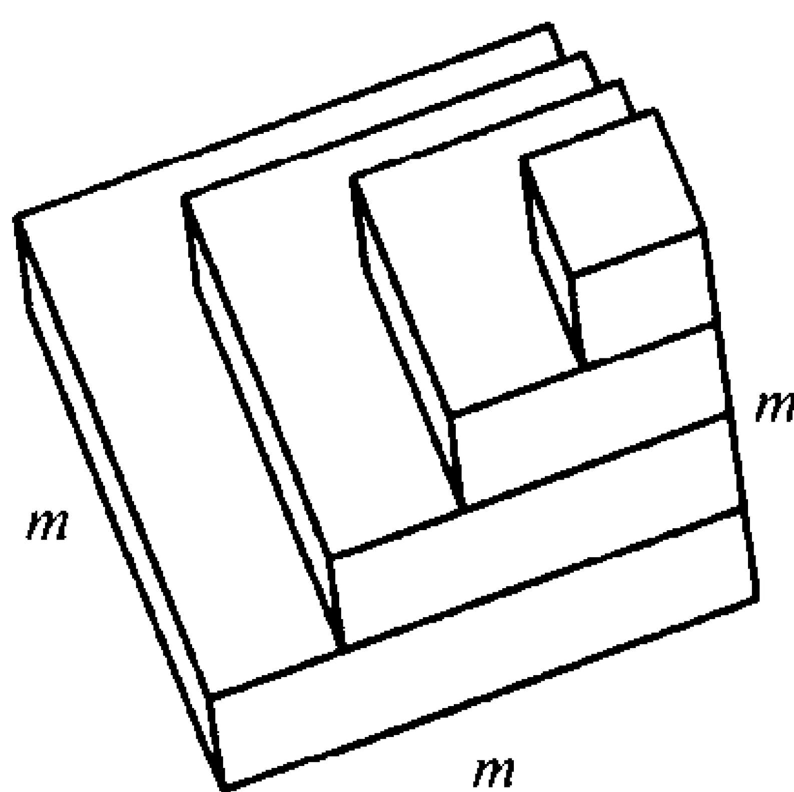
$U = A, L = I$
for $k = 1$ **to** $m-1$
　　for $j = k+1$ **to** m
　　　　$\ell_{jk} = u_{jk}/u_{kk}$
　　　　$u_{j,k:m} = u_{j,k:m} - \ell_{jk} u_{k,k:m}$

(3 个矩阵 $\boldsymbol{A}$，$\boldsymbol{L}$，$\boldsymbol{U}$ 并非实际的需要；为使计算机存储最小化，$\boldsymbol{L}$ 和 $\boldsymbol{U}$ 两者都可以写进与 $\boldsymbol{A}$ 相同的数组中.）作为高斯消元法的另一个“外积”公式，参看习题 20.4，那里包含了一个而非两个显式的循环.

20.4　运算计数

如通常那样，这个算法的渐近运算计数可以几何地导出. 工作量主要取决于在内循环的向量运算，$u_{j,k:m} = u_{j,k:m} - \ell_{jk} u_{k,k:m}$，它执行了一个数量-向量乘法和一个向量减法. 若用 $\ell = m - k + 1$ 表示所处理的行向量长度，flop 的数目是 2ℓ：每个元素 2 次 flop.

对每个 k 值，内循环对第 $k+1,\cdots,m$ 行重复. 所含的工作量对应于下面几何体的一层： 151



这个图形和我们在第 10 讲中表示豪斯霍尔德三角形化（设 $m = n$）工作量所显示的图形是相同的. 然而，在那里每个单位立方体表示 4 次 flop 而不是 2 次. 如前，当 $m \to \infty$ 时，立方体收敛到角锥，其体积为 $\frac{1}{3}m^3$. 在每单位体积 2 次 flop 的情况下，总计为

$$\text{高斯消元法的工作量：} \sim \frac{2}{3}m^3 \text{ 次 flop.} \tag{20.8}$$

20.5　用 LU 因子分解解 $Ax=b$

如果将 A 因子分解为 L 和 U，那么方程组 $Ax=b$ 化为 $LUx=b$ 的形式，则该方程组可通过解两个三角形方程组来求解：首先对未知的 y 解 $Ly=b$（向前代入），然后对未知的 x 解 $Ux=y$（回代）. 第一步需要 $\sim\frac{2}{3}m^3$ 次 flop，第二和第三步各需 $\sim m^2$ 次 flop. 总的工作量为 $\sim\frac{2}{3}m^2$ 次 flop，这是用豪斯霍尔德三角形化求解（算法 16.1）的次数 $\sim\frac{4}{3}m^3$（10.9）的一半.

为什么解正方形的方程组通常用高斯消元法而不用 QR 因子分解？因子 2 显然是一个理由，然而，同样重要的是这样一个历史事实，消元法的思想已经被人知晓了若干世纪，而直到计算机发明以后矩阵的 QR 因子分解才得以出现. 作为取代高斯消元法的方法，QR 因子分解一定会有令人信服的优势.

20.6　不选主元素的高斯消元法的不稳定性

可惜的是，到目前为止，所给出的高斯消元法不能用于解一般的线性方程组，因为它不是向后稳定的. 不稳定性涉及另外一个更加明显的困难. 对某些矩阵，高斯消元法因为要企图除以零而导致完全失败.

例如，考虑矩阵

$$A=\begin{bmatrix}0 & 1\\ 1 & 1\end{bmatrix}.$$

152

这个矩阵是满秩且是良态的，按 2-范数，$\kappa(A)=(3+\sqrt{5})/2\approx 2.618$. 然而，高斯消元法在第一步就失败了.

同样的矩阵一个轻微的扰动揭示了更一般的问题. 假设将高斯消元法应用到

$$A=\begin{bmatrix}10^{-20} & 1\\ 1 & 1\end{bmatrix}. \tag{20.9}$$

现在求解过程并未失败. 相反，第二行减去 10^{20} 乘第一行，产生下面的因子：

$$L=\begin{bmatrix}1 & 0\\ 10^{20} & 1\end{bmatrix},\qquad U=\begin{bmatrix}10^{-20} & 1\\ 0 & 1-10^{20}\end{bmatrix}.$$

然而，假设在 $\epsilon_{\text{机器}}\approx 10^{-16}$ 的浮点运算中完成这些计算. 数 $1-10^{20}$ 并不会被准确地表

示，它要舍入到最接近的浮点数. 为了简单起见，想像它是准确的 -10^{20}，则由算法产生的浮点矩阵将是

$$\tilde{L}=\begin{bmatrix}1 & 0\\ 10^{20} & 1\end{bmatrix},\qquad \tilde{U}=\begin{bmatrix}10^{-20} & 1\\ 0 & -10^{20}\end{bmatrix}.$$

最初看来舍入的程度似乎是可以容许的，毕竟矩阵 $\tilde{U}$ 相对于 $\|U\|$ 来说接近 U，然而，当我们计算乘积 $\tilde{L}\tilde{U}$：

$$\tilde{L}\tilde{U}=\begin{bmatrix}10^{-20} & 1\\ 1 & 0\end{bmatrix},$$

问题就出现了. 由于在（2，2）的位置上 1 以 0 代替了，所以这个矩阵根本不接近 A. 如果我们现在解方程组 $\tilde{L}\tilde{U}x=b$，结果将不是 $Ax=b$ 的解. 例如，$b=(1,0)^*$，我们得到 $\tilde{x}=(0.1)^*$，而正确的解是 $x\approx(-1,1)^*$.

下面将详细解释在这个例子中所出现的情况. 高斯消元法稳定地计算 LU 因子分解：对于接近 A 的矩阵（事实上是 A 自己），$\tilde{L}$ 和 $\tilde{U}$ 接近准确的因子，然而解 $Ax=b$ 并不稳定. 解释是 LU 因子分解虽然稳定，但*不是向后稳定的*. 把它总结为一个法则，如果算法中的一步对解一个子问题的稳定但不向后稳定，那么全部计算的稳定性可能处于危险之中.

事实上，对一般的 $m\times m$ 矩阵 A，情况比这更糟糕. 不选主元素的高斯消元法作为一个一般的 LU 因子分解算法既不向后稳定也不稳定. 另外，所产生的三角形矩阵的条件数，可能任意大于 A 本身的条件数，这将是导致在解 $Ax=b$ 的向前代入和回代中产生附加的不稳定性来源. 153

习　题

20.1 令 $A\in\mathbb{C}^{m\times m}$ 非奇异. 证明 A 有 LU 因子分解当且仅当对每个满足 $1\leqslant k\leqslant m$ 的 k，左上角的 $k\times k$ 块 $A_{1:k,1:k}$ 非奇异.（提示：高斯消元法留下的行列式 det（$A_{1:k,1:k}$）不变.）证明这个 LU 因子分解是惟一的.

20.2 设 $A\in\mathbb{C}^{m\times m}$ 满足习题 20.1 的条件，且是带宽为 $2p+1$ 的带状矩阵，即对 $|i-j|>p$，有 $a_{ij}=0$，A 的因子 L 和 U 有什么样的稀疏形式？

20.3 设 $m\times m$ 矩阵 A 写成块状形式 $A=\begin{bmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix}$，其中 A_{11} 是 $n\times n$ 的，A_{22} 是 $(m-n)\times(m-n)$ 的. 假设 A 满足习题 20.1 的条件.

(a) 验证“消去”块 A_{21} 的公式

$$\begin{bmatrix} I & \\ -A_{21}A_{11}^{-1} & I \end{bmatrix}\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix}.$$

矩阵 $A_{22} - A_{21}A_{11}^{-1}A_{12}$ 称为 A_{11} 在 A 的舒尔余（Schur complement）.

(b) 假设 A_{21} 借助高斯消元法的 n 步逐行被消去. 证明结果右下方的 $(m-n)\times(m-n)$ 块再次得到 $A_{22} - A_{21}A_{11}^{-1}A_{12}$.

20.4 像本书大多数的算法那样，高斯消元法包含了 3 重嵌套循环. 在算法 10.1，有两个显式的 **for** 循环，第 3 个循环隐含在向量 $u_{j,k:m}$ 和 $u_{k,k:m}$ 中. 用只有一个指标为 k 的显式 **for** 循环重写这个算法. 在这个循环中，U 在每步以某个秩 1 外积作更新. 如果要想优化计算机性能的话，这种高斯消元法的“外积”形式可能是比算法 20.1 更好的开始点.

20.5 我们已经看到高斯消元法产生了因子分解 $A = LU$，其中 L 在对角线有元素 1，而 U 并非如此. 如果这个过程以下面的方式变化，比较概念性地描述所得到的因子分解：

(a) 由左到右按列消元，而不是由顶部到底部按行消元，这样 A 成为下三角的形式.

(b) 在事先以对角矩阵 D 按比例改变 A 的列之后使用高斯消元法，方程组 $Ax=b$ 在这个更改比例的情况下取什么形式? 方程组或未知数也由 D 更改比例吗?

(c) 高斯消元法带来更多的东西，使得 A（设为非奇异）产生上三角形形式之后，附加的列运算使得这个上三角形矩阵成为对角矩阵.

154

第21讲 选 主 元

在前一讲中，我们看到纯粹形式的高斯消元法是不稳定的. 不稳定性可以由置换运算中矩阵的行次序来控制，这是一种被称为选主元(pivoting)的运算. 自从20世纪50年代以来，选主元已经成为高斯消元法计算的标准成分.

21.1 主 元

在高斯消元法的第 k 步，为了在运算中的矩阵 X 的第 $k+1,\cdots,m$ 行的第 k 个元素引入零，由这些行减去第 k 行的倍数. 在运算中第 k 行，第 k 列，特别是，元素 x_{kk} 扮演了特殊的角色. 我们称 x_{kk} 为主元(pivot). 子矩阵 $X_{k+1:m,k:m}$ 的每个元素减去第 k 行一个数和第 k 列一个数的乘积除以 x_{kk}：

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}.$$

然而，为了消元没有理由一定要选择第 k 行和第 k 列. 例如，可能以某个第 i 行 $(k<i\leqslant m)$ 的倍数加到其他第 $k,\cdots,m$ 行，这样刚好容易在第 k 列引入零. 在此情形中，元素 x_{ik} 将是主元. 这里给出 $k=2$ 和 $i=4$ 的一个例子：

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & x_{ik} & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}.$$

类似地，我们可以在第 j 列而不是第 k 列引入零. 这里是 $k=2$，$i=4$，$j=3$ 的一个例子：

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \boldsymbol{\times} & x_{ij} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & \boldsymbol{\times} & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & x_{ij} & \times & \times \\ & \boldsymbol{\times} & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}.$$

总的来说，我们可自由地选择 $X_{k:m,k:m}$ 的任意元素作为主元，只要它非零. 元素 $x_{kk}=0$

的可能性隐含了主元选择的某些灵活性通常是必需的，甚至从纯数学的观点来看也是这样．然而，对于数值稳定性，甚至在 x_{kk} 非零时，如果可以得到一个较大的元素，选主元也是值得的．实际上，通常在一个被考虑为候选者的元素所组成的集合中取最大的数作为主元．

如果在整个矩阵以任意方式引入零，消元过程的结构会很快引起混乱．为了清楚如何进行，我们会希望保持上一讲描述的三角形结构，而且也容易做到这一点．我们不想主元像上面所说的那样留在原处．代之的是，在第 k 步我们想像运算中矩阵的行和列被置换以便把 x_{ij} 移动到 (k,k) 位置上．这样，当消元完成时，就在第 k 列的第 $k+1,\cdots,m$ 个元素引入了零，如同在不选主元的高斯消元法中那样．这种行或列的交换就是通常*选主元*(pivoting)的想法．

交换行和列的想法在概念上是避免不了的，在计算机上实际交换它们是否是一个好想法还不很清楚．在某些实现中，计算机存储的数据在每个选主元步骤上真正地进行交换．在其他情形，可由置换变址向量间接寻址来达到等价的效果．哪一种方法最好因机器的不同而异，并取决于其他很多因素．

21.2　部分选主元

如果在第 k 步考虑 $\boldsymbol{X}_{k:m,k:m}$ 的每个元素都可能作为主元，则要检验 $O((m-k)^2)$ 个元素以确定最大者．所有 m 步加起来，选主元全部的代价变成 $O(m^3)$ 次运算，这大大地增加了高斯消元法的代价，且不说以一种不能预见的方式通过矩阵所有元素的全局交换所带来的潜在的困难．这种昂贵的策略称为*完全选主元*（complete pivoting）．

156

实际上，可以考虑在少得多的元素中找到同样好的主元．这样做的标准方法是*部分选主元*(partial pivoting)．这里，只有行被交换．每步的主元选为第 k 列的 $m-k+1$ 个对角线以下元素的最大者，每步选主元的全部代价只有 $O(m-k)$ 次运算，因而总的运算量是 $O(m^2)$，将第 k 个主元放在 (k,k) 位置，不需要置换列，只要将第 k 行与含有主元的行交换就可以了．

$$\underset{\text{选主元}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{P_1} \underset{\text{行交换}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{L_1} \underset{\text{消元}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}.$$

像通常数值线性代数的算法那样，这个算法可以表示为矩阵的乘积．在上一讲中看到，一个消元步对应左乘以一个初等下三角形矩阵 $\boldsymbol{L}_k$．部分选主元在每次消元之前先将置换矩阵 $\boldsymbol{P}_k$ 用在正在运算的矩阵的左边，使事情复杂了．（一个置换矩阵

是每行和每列除了有单个的元素 1 外，其他各处都为 0 的矩阵. 即它是由单位矩阵置换行或列得到的矩阵.）$m-1$ 步之后，$\boldsymbol{A}$ 变成一个上三角阵 $\boldsymbol{U}$：

$$\boldsymbol{L}_{m-1}\boldsymbol{P}_{m-1}\cdots\boldsymbol{L}_2\boldsymbol{P}_2\boldsymbol{L}_1\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U}. \tag{21.1}$$

21.3　例　　子

回到上一讲的例子（20.3），这有助于观察过程如何进行，

$$\boldsymbol{A}=\begin{bmatrix}2&1&1&0\\4&3&3&1\\8&7&9&5\\6&7&9&8\end{bmatrix}. \tag{21.2}$$

157

用部分选主元，首先是交换第 1 行和第 3 行（左乘以 $\boldsymbol{P}_1$）：

$$\begin{bmatrix}&&1&\\&1&&\\1&&&\\&&&1\end{bmatrix}\begin{bmatrix}2&1&1&0\\4&3&3&1\\8&7&9&5\\6&7&9&8\end{bmatrix}=\begin{bmatrix}8&7&9&5\\4&3&3&1\\2&1&1&0\\6&7&9&8\end{bmatrix}.$$

第 1 步消元就像是（左乘以 $\boldsymbol{L}_1$）：

$$\begin{bmatrix}1&&&\\-\frac{1}{2}&1&&\\-\frac{1}{4}&&1&\\-\frac{3}{4}&&&1\end{bmatrix}\begin{bmatrix}8&7&9&5\\4&3&3&1\\2&1&1&0\\6&7&9&8\end{bmatrix}=\begin{bmatrix}8&7&9&5\\&-\frac{1}{2}&-\frac{3}{2}&-\frac{3}{2}\\&-\frac{3}{4}&-\frac{5}{4}&-\frac{5}{4}\\&\frac{7}{4}&\frac{9}{4}&\frac{17}{4}\end{bmatrix}.$$

现在交换第 2 行和第 4 行（乘以 $\boldsymbol{P}_2$）：

$$\begin{bmatrix}1&&&\\&&&1\\&&1&\\&1&&\end{bmatrix}\begin{bmatrix}8&7&9&5\\&-\frac{1}{2}&-\frac{3}{2}&-\frac{3}{2}\\&-\frac{3}{4}&-\frac{5}{4}&-\frac{5}{4}\\&\frac{7}{4}&\frac{9}{4}&\frac{17}{4}\end{bmatrix}=\begin{bmatrix}8&7&9&5\\&\frac{7}{4}&\frac{9}{4}&\frac{17}{4}\\&-\frac{3}{4}&-\frac{5}{4}&-\frac{5}{4}\\&-\frac{1}{2}&-\frac{3}{2}&-\frac{3}{2}\end{bmatrix}.$$

第 2 步消元则像是（乘以 $\boldsymbol{L}_2$）：

$$\begin{bmatrix}1 & & & \\ & 1 & & \\ & \frac{3}{7} & 1 & \\ & \frac{2}{7} & & 1\end{bmatrix}\begin{bmatrix}8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2}\end{bmatrix}=\begin{bmatrix}8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7}\end{bmatrix}.$$

现在交换第 3 行和第 4 行（乘以 $\boldsymbol{P}_3$）：

$$\begin{bmatrix}1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix}\begin{bmatrix}8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7}\end{bmatrix}=\begin{bmatrix}8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & -\frac{2}{7} & \frac{4}{7}\end{bmatrix}.$$

最后的消元步就像是（乘以 $\boldsymbol{L}_3$）：

$$\begin{bmatrix}1 & & & \\ & 1 & & \\ & & 1 & \\ & & -\frac{1}{3} & 1\end{bmatrix}\begin{bmatrix}8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & -\frac{2}{7} & \frac{4}{7}\end{bmatrix}=\begin{bmatrix}8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3}\end{bmatrix}.$$

158

21.4 $PA = LU$ 因子分解和第 3 个幸运之处

我们刚才已经计算了 $\boldsymbol{A}$ 的 LU 因子分解了吗？不完全是，但已经几乎是了．事实上，我们计算了 $\boldsymbol{PA}$ 的 LU 因子分解，其中 $\boldsymbol{P}$ 是一个置换矩阵．它就是：

$$\underbrace{\begin{bmatrix} & & 1 & \\ & & & 1 \\ & 1 & & \\ 1 & & & \end{bmatrix}}_{P}\underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}}_{A}=\underbrace{\begin{bmatrix} 1 & & & \\ \frac{3}{4} & 1 & & \\ \frac{1}{2} & -\frac{2}{7} & 1 & \\ \frac{1}{4} & -\frac{3}{7} & \frac{1}{3} & 1 \end{bmatrix}}_{L}\underbrace{\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}}_{U}. \quad (21.3)$$

这个公式可以和（20.5）作比较. 那里给出的整数和这里的分数没有一般的区别，只是 A 的选择的人为结果. 不同点在于，这里 L 的所有对角线以下的元素按绝对值都小于等于 1，这是引入选主元后性质 $|x_{kk}| = \max_j |x_{jk}|$ 在（20.6）所得的结论.

（21.3）如何得到并不是显然的，消去过程取

$$L_3P_3L_2P_2L_1P_1A = U,$$

的形式，它根本看不出下三角形形式. 但在这里，第 3 个好运气帮助了我们. 这 6 个初等运算可以重新调整次序为以下形式

$$L_3P_3L_2P_2L_1P_1 = L'_3L'_2L'_1P_3P_2P_1, \quad (21.4)$$

其中 L'_k 等于 L_k，但是对角线以下的元素需要被置换. 其精确定义为

$$L'_3 = L_3, L'_2 = P_3L_2P_3^{-1}, L'_1 = P_3P_2L_1P_2^{-1}P_3^{-1}.$$

因为这些定义中的每一个只用了置换 $P_j(j>k)$ 到 L_k，容易验证 L'_k 和 L_k 有同样的结构. 计算矩阵 L'_k 的乘积可得

$$L'_3L'_2L'_1P_3P_2P_1 = L_3(P_3L_2P_3^{-1})(P_3P_2L_1P_2^{-1}P_3^{-1})P_3P_2P_1 = L_3P_3L_2P_2L_1P_1,$$

如（21.4）所示.

一般地，对于 $m \times m$ 矩阵，由部分选主元的高斯消元法提供的因子分解（21.1）可写成

$$(L'_{m-1}\cdots L'_2L'_1)(P_{m-1}\cdots P_2P_1)A = U, \quad (21.5)$$

其中 L'_k 定义为

$$L'_k = P_{m-1}\cdots P_{k+1}L_kP_{k+1}^{-1}\cdots P_{m-1}^{-1}. \quad (21.6)$$

159

矩阵$\{L'_k\}$和乘积一定是单位下三角形阵，且容易由取其对角线下元素的相反数来求逆，正如在不选主元的高斯消元法中那样. 记 $L = (L'_{m-1}\cdots L'_2L'_1)^{-1}$ 和 $P = P_{m-1}\cdots P_2P_1$，得到

$$PA = LU. \tag{21.7}$$

一般地，任何一个奇异或非奇异的正方形矩阵都有因子分解（21.7），其中 $\boldsymbol{P}$ 是一个置换矩阵，$\boldsymbol{L}$ 是下三角元素按绝对值小于等于 1 的单位下三角形矩阵，$\boldsymbol{U}$ 是上三角形矩阵. 部分选主元是如此一种通用的实践方法，以致于这个因子分解通常简称为 $\boldsymbol{A}$ 的 *LU 因子分解*（LU factorization）.

著名的公式（21.7）有一个简单的解释，部分选主元的高斯消元等价于下列过程：

1. 根据 $\boldsymbol{P}$ 置换 $\boldsymbol{A}$ 的行.
2. 对 $\boldsymbol{PA}$ 应用不选主元的高斯消元法.

当然，因为 $\boldsymbol{P}$ 并不在事前知道，所以实际上部分选主元并不以这样的方式进行.

这里是此算法的正式陈述.

算法 21.1 部分选主元的高斯消元法

$U = A$，$L = I$，$P = I$
for $k = 1$ **to** $m-1$
  选 $i \geq k$ 使 $|u_{ik}|$ 最大化
  $u_{k,k:m} \leftrightarrow u_{i,k:m}$（交换两行）
  $\ell_{k,1:k-1} \leftrightarrow \ell_{i,1:k-1}$
  $p_{k,:} \leftrightarrow p_{i,:}$
  for $j = k+1$ **to** m
    $\ell_{jk} = u_{jk}/u_{kk}$
    $u_{j,k:m} = u_{j,k:m} - \ell_{jk} u_{k,k:m}$

为了调整次序，算法需要和不选主元的高斯消元法同样数目的浮点运算（20.8），即 $\frac{2}{3}m^3$. 如同在算法 20.1 中一样，如果将 $\boldsymbol{U}$ 和 $\boldsymbol{L}$ 全部写入存储 $\boldsymbol{A}$ 的同样的数组，则可以最小化计算机的存储.

当然，实际上 $\boldsymbol{P}$ 并不能显式地表示为一个矩阵. 算法的每一步都要调动行，或借助一个置换向量达到等价的效果，如早先指出的那样.

160

21.5 完全选主元

在完全选主元方法中，主元的选择总共要花费相当多的时间，实际上是很少这样做的，因为稳定性的改进是有限的. 然而，我们还是概述一下在这种情形下代数的变动.

以矩阵的形式描述，完全选主元在每个消元步骤之前以一个行的置换 $\boldsymbol{P}_k$ 用在左边，同时以一个列的置换 $\boldsymbol{Q}_k$ 用在右边：

$$L_{m-1}P_{m-1}\cdots L_2P_2L_1P_1AQ_1Q_2\cdots Q_{m-1}=U. \tag{21.8}$$

同样，这并不完全是A的一个LU因子分解，但它基本上是. 如果L'_k如同（21.6）那样定义（不包含列的置换），则

$$(L'_{m-1}\cdots L'_2L'_1)(P_{m-1}\cdots P_2P_1)A(Q_1Q_2\cdots Q_{m-1})=U. \tag{21.9}$$

令$L=(L'_{m-1}\cdots L'_2L'_1)^{-1}$，$P=P_{m-1}\cdots P_2P_1$，和$Q=Q_1Q_2\cdots Q_{m-1}$，得到

$$PAQ=LU. \tag{21.10}$$

习 题

21.1 令A为本讲和前一讲中所考虑的4×4矩阵（20.3）.

(a) 由（20.5）求$\det A$.

(b) 由（21.3）求$\det A$.

(c) 描述部分选主元的高斯消元法如何用于求一个一般正方形矩阵的行列式.

21.2 假设$A\in\mathbb{C}^{m\times m}$是习题20.2的带宽为$2p+1$的带状矩阵，及由部分选主元的高斯消元法计算出的因子分解$PA=LU$，描述关于L和U的稀疏方式.

21.3 考虑由列代替行作出的高斯消元法，导出因子分解$AQ=LU$，其中Q是一个置换矩阵.

(a) 证明若A非奇异，这样的因子分解总存在.

(b) 证明若A奇异，这样的因子分解不总存在.

21.4 高斯消元法可以用于计算非奇异矩阵$A\in\mathbb{C}^{m\times m}$的逆$A^{-1}$，虽然很少真的这样做.

(a) 描述由解m个方程组计算A^{-1}的算法，并指出它的渐近运算计数为$8m^3/3$次flop. 161

(b) 描述你的算法的一个变形，取稀疏性的优点，化运算计数为$2m^3$次flop.

(c) 假设要解n个方程组$Ax_j=b_j$，或等价地解一个块方程组$AX=B$，其中$B\in\mathbb{C}^{m\times n}$. 下面各自的渐近运算计数（作为$m$和$n$的函数）是什么？（i）直接由LU因子分解（ii）用预先计算A^{-1}.

21.5 假设$A\in\mathbb{C}^{m\times m}$是埃米尔特矩阵，或在实的情形是对称矩阵（但不必是正定的）.

(a) 描述一个保持埃米尔特结构的对称主元策略，它仍然导出一个元素$|\ell_{ij}|\leq1$的单位下三角矩阵.

(b) 由你的算法计算的矩阵因子分解是什么形式？

(c) 渐近运算计数是什么？

21.6 假设$A\in\mathbb{C}^{m\times m}$是严格列对角占优的（strictly column diagonally dominant），意思是对每个k，

$$|a_{kk}|>\sum_{j\neq k}|a_{jk}|. \tag{21.11}$$

证明如果部分选主元的高斯消元法用于A，则不发生行交换.

21.7 在第20讲中用向量e_k和ℓ_k解释了“两个幸运之处”. 对本讲中“第三个幸运之处”给出一个基于这些向量的一个说明. 162

第 22 讲　高斯消元法的稳定性

部分选主元的高斯消元法对某些矩阵理论上讲是极不稳定的，但在实际上却是稳定的．这个表面的悖论有其统计意义下的解释．

22.1　稳定性和 L 与 U 的大小

数值线性代数大多数算法包括所有基于酉运算的算法的稳定性分析是直接的．然而，部分选主元的高斯消元法的稳定性分析是复杂的，自从 20 世纪 50 年代以来这就是数值分析的一个难点．这是在本书中我们到后半部分才讨论高斯消元法的理由之一．

在（20.9）中给出了不选主元的高斯消元法不稳定的一个 2×2 矩阵的例子．在那个例子中，因子 L 有一个 10^{20} 大小的元素．一个基于 L 解方程组的尝试引入相对阶 $\epsilon_{机器}$ 的舍入误差，因而绝对阶为 $\epsilon_{机器}\times 10^{20}$．这不足为奇，它破坏了结果的准确度．

这个例子在某种意义下完全是一般的．选或不选主元的高斯消元法，如果因子 L 和 U 的一者或两者相对于 A 的大小来说是大的，只有这种情况下能产生不稳定性．因 163 此从稳定性的观点来看，选主元的目的要保证 L 和 U 不太大．只要发生在消元过程中的所有中间量的大小可以控制，它们忽略的舍入误差是非常小的，且算法向后稳定．

下面的定理精确地描述了这个思想．它是对不选主元的高斯消元法建立的，但是如果 A 表示为原矩阵适当置换了行和/或列所得的矩阵，定理应用到选主元的消元法也成立．

定理 22.1　令非奇异矩阵 $A\in\mathbb{C}^{m\times m}$ 的因子分解 $A=LU$ 由不选主元的高斯消元法在一个满足公理（13.5）和（13.7）的计算机上计算．如果 A 有一个 LU 因子分解，则对所有充分小的 $\epsilon_{机器}$，因子分解以浮点运算成功地完成（没有遇到零主元），且计算出来的矩阵 $\tilde{L}$ 和 $\tilde{U}$ 对某些 $\delta A\in\mathbb{C}^{m\times m}$ 满足

$$\tilde{L}\tilde{U}=A+\delta A,\qquad \frac{\|\delta A\|}{\|L\|\|U\|}=O(\epsilon_{机器}).\tag{22.1}$$

通常在数值线性代数中，对 $\tilde{L}-L$ 或 $\tilde{U}-U$ 不作什么要求，仅对 $\tilde{L}\tilde{U}-LU$ 作要求．

乍看起来这个估计就像本书中很多其他的估计，例如（16.3）或（17.3）那样．它的不同之处在于分母的量是 $\|L\|\|U\|$ 而不是 $\|A\|$．如果 $\|L\|\|U\|=O(\|A\|)$，则（22.1）断定高斯消元法向后稳定．然而，如果 $\|L\|\|U\|\neq O(\|A\|)$，我们一定会预期到向后不稳定．

对不选主元的高斯消元法，L 和 U 两者可以无界地大，算法以任何标准看都是不稳定的，我们将不进一步讨论．代之的是，从现在开始我们把注意力放在部分选主元的高斯消元法上．

22.2 增长因子

考虑部分选主元的高斯消元法. 因为每一次主元的选择包含了在一列上的最大化，这个算法产生了矩阵 L，它在对角线以下各处元素的绝对值都小于等于 1，这隐含了对任何范数都有 $\|L\| = O(1)$. 因此，对选主元的高斯消元法，(22.1) 化为条件 $\|\delta A\|/\|U\| = O(\epsilon_{机器})$. 我们断定只要 $\|U\| = O(\|A\|)$，算法就是向后稳定的.

这个结论有一个标准的重新阐述的形式，它或许是更清晰的. 高斯消元法把一个满矩阵 A 约化成一个上三角形矩阵 U. 刚才已经看到，对稳定性的关键问题是在这种约化过程中元素是否被放大. 实际上，定义 A 的增长因子(growth factor)为如下的比

$$\rho = \frac{\max_{i,j}|u_{ij}|}{\max_{i,j}|a_{ij}|}. \tag{22.2}$$

164

如果 ρ 为阶 1，不会发生多大的增长，消元过程是稳定的. 如果 ρ 比这更大，我们一定可以预期不稳定. 特别地，因为 $\|L\| = O(1)$，且 (22.2) 隐含 $\|U\| = O(\rho\|A\|)$，以下的结论是定理 22.1 的一个推论.

定理 22.2 令矩阵 $A \in \mathbb{C}^{m\times m}$ 的因子分解 $PA = LU$ 由部分选主元的高斯消元法 (算法 21.1) 在一台满足公理 (13.5) 和 (13.7) 的计算机上计算，则计算出的矩阵 $\tilde{P}$，$\tilde{L}$ 和 $\tilde{U}$ 对某些 $\delta A \in \mathbb{C}^{m\times m}$ 满足

$$\tilde{L}\tilde{U} = \tilde{P}A + \delta A, \qquad \frac{\|\delta A\|}{\|A\|} = O(\rho\,\epsilon_{机器}). \tag{22.3}$$

其中 ρ 为 A 的增长因子. 如果对每个 $i>j$ 有 $|\ell_{ij}|<1$，这隐含了在精确的运算中，主元的选择不存在几个元素同时出现的现象，那么对所有充分小的 $\epsilon_{机器}$，有 $\tilde{P} = P$.

高斯消元法是否向后稳定？根据定理 22.2 和向后稳定的定义 (14.5)，答案是：如果对给定维数 m 的所有矩阵 $\rho = O(1)$ 一致地成立，则向后稳定，否则不向后稳定.

现在，复杂的情况开始出现了.

22.3 最坏情形的不稳定性

对某些矩阵 A，尽管有选主元的有利效果，但是 ρ 还是可能会是巨大的. 例如，假设 A 是矩阵

$$A = \begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}. \tag{22.4}$$

在第一步，没有发生选主元，但最后一列的第 $2,3,\cdots,m$ 个元素由 1 到 2 变成了两倍. 另外的加倍出现在每个子序列的消元步. 最后有

$$
U=\begin{bmatrix}1&&&&1\\&1&&&2\\&&1&&4\\&&&1&8\\&&&&16\end{bmatrix}. \tag{22.5}
$$

165

最后 $PA=LU$ 因子分解为:

$$
\begin{bmatrix}1&&&&1\\-1&1&&&1\\-1&-1&1&&1\\-1&-1&-1&1&1\\-1&-1&-1&-1&1\end{bmatrix}=\begin{bmatrix}1&&&&\\-1&1&&&\\-1&-1&1&&\\-1&-1&-1&1&\\-1&-1&-1&-1&1\end{bmatrix}\begin{bmatrix}1&&&&1\\&1&&&2\\&&1&&4\\&&&1&8\\&&&&16\end{bmatrix}.
$$

对此 5×5 矩阵，增长因子为 $\rho=16$. 对同样形式的 $m\times m$ 矩阵，有 $\rho=2^{m-1}$. （这是 ρ 可取到的最大值，参看习题 22.1.）

2^m 阶的增长因子对应于精度为 m 个 bit 阶的损失，这对实际计算是一个灾难. 因为典型的计算机刚好用 64 个 bit 表示浮点数，而维数为数百，数千的矩阵问题一直都在计算机上求解，m 个 bit 的精度损失对实际的计算是不能允许的.

这给我们带来一个难点. 这里，在选主元的高斯消元法的讨论中（在本书只有这一次）第 14 讲给出的稳定性定义对我们来说失败了. 根据定义，确定稳定或向后稳定的所有事情是一个确定的界的存在性问题，这个界应是能够一致地应用于每个固定维数 m 的所有矩阵的，对 m 的一致性是不需要的. 这里，对每个 m，有一个包含常数 2^{m-1} 的一致界，因此根据我们的定义，高斯消元法向后稳定.

定理 22.3 根据第 14 讲的定义，部分选主元的高斯消元法向后稳定.

然而，鉴于 2^{m-1} 在实际 m 值下是巨大的，这个结论并不合理.

对于本讲的剩余部分，我们要求读者将稳定的正规定义放在一边，接受一个非正式（及更标准）的用词. 高斯消元法对某些矩阵是极其不稳定的，这可以用 MATLAB，LINPACK，LAPACK 或其他有极好声誉的软件包的数值实验来证实.

22.4 实际上的稳定性

如果高斯消元法不稳定，为什么它却如此著名和如此普及? 这给我们带来一点启示，即不能仅考虑一个人为的定义，而要考虑关于这个算法实际的性质. 尽管有像（22.4）那样的例子，部分选主元的高斯消元法实际上是完全稳定的. 像（22.5）那样的大因子 U 似乎永不出现在真实的应用中. 在 50 多年来的计算实践中，在自然环境下从没有发生极不稳定的矩阵问题.

166

这确实是一个奇特的情况. 对某些矩阵失败的算法却能够在实际上完全值得信赖. 虽然一些矩阵引起不稳定性，但这些代表了所有矩阵的集合中相当小的比例，简单地，由于统计的理由，它们在实际上“永不”发生.

可以进一步考虑用随机矩阵来研究这种现象. 确实，在应用中产生的矩阵在任何通常意义下是非随机的. 它们具备所有的特殊性质，如果有人试图将它们描述为某个分布的随机样本，那将不得不是某一个奇特的分布. 显然没有理由期待，随机矩阵存在一个特殊的分布会以一个接近量化的途径符合在实际中发生的矩阵的行为.

然而，要解释的现象不是精确定量的. 在应用中具有大增长因子的矩阵几乎近似为零. 如果能够表明它们在某些明确定义的随机矩阵类中是几乎近似为零，那么相关的机理也肯定是相同的. 这并不依赖于如何衡量“近似为零”，无论是因子 2 还是 10 还是 100.

图 22-1 和图 22-2 给出如习题 12.3 定义的随机矩阵的实验：每一个元素是平均值为 0，标准差为 $m^{-1/2}$ 的实正态分布的独立样本. 在图 22-1 中，各种维数的随机矩阵族被因子分解，且增长因子用散点表示. 这些矩阵中只有两个矩阵给出了大于 $m^{1/2}$ 的增长因子. 在图 22-2 中，显示了维数为 $m=8$，16 和 32 的各一百万个矩阵因子分解的结果. 这里，增长因子已经收集在宽度为 0.2 的框架内，得出的数据作为概率密度分布画在图上. 增长因子的概率密度呈现出对其大小按指数减少. 在这 300 万个矩阵中，虽然原则上最大增长因子可以是 2, 147, 483, 648，实际上碰到的最大值是 11.99.

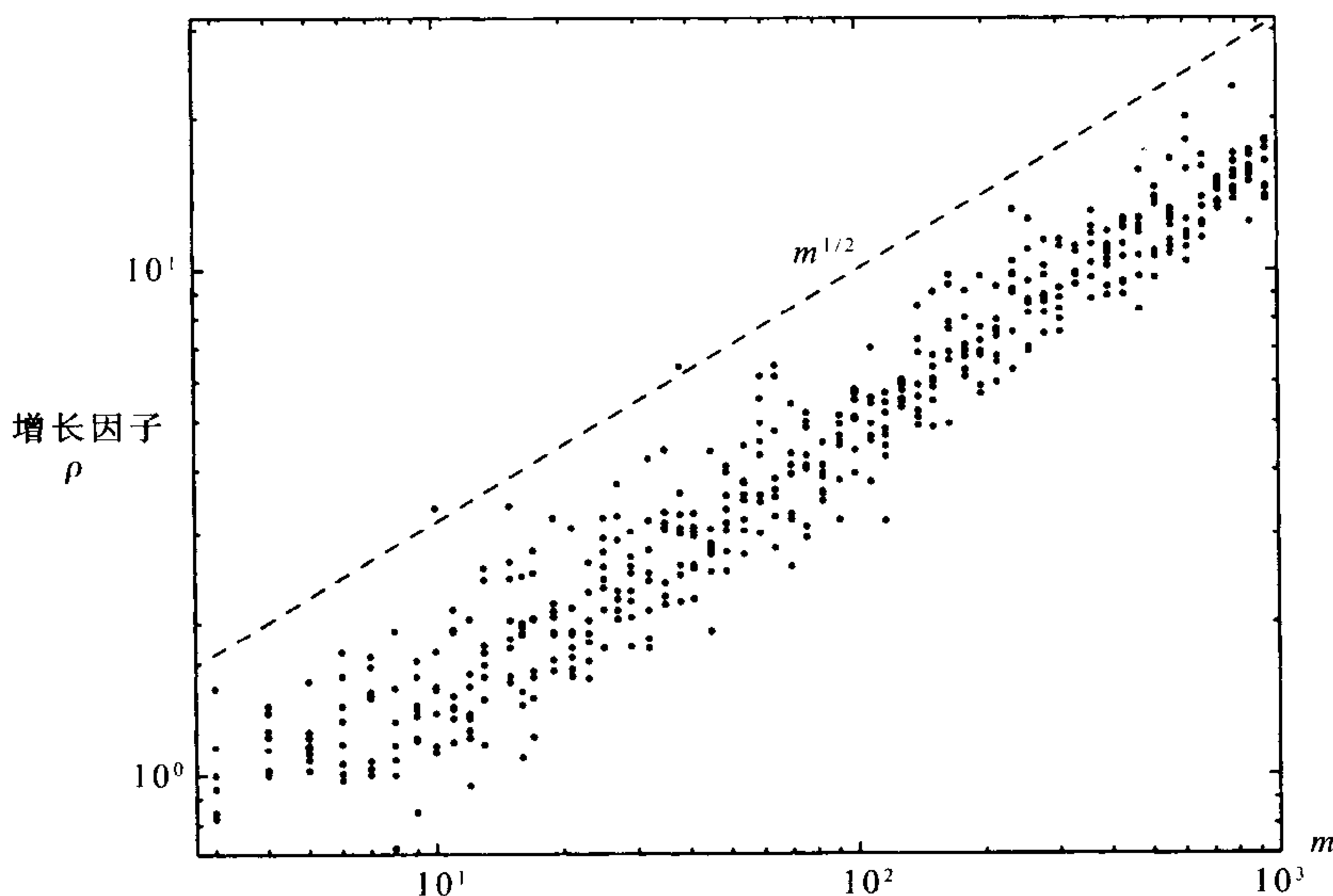


图 22-1　选主元高斯消元法应用到各种维数的 496 个随机矩阵（独立的，正态分布元素）的增长因子. ρ 的典型大小是 $m^{1/2}$ 阶的，大大小于最大可能的值 2^{m-1}

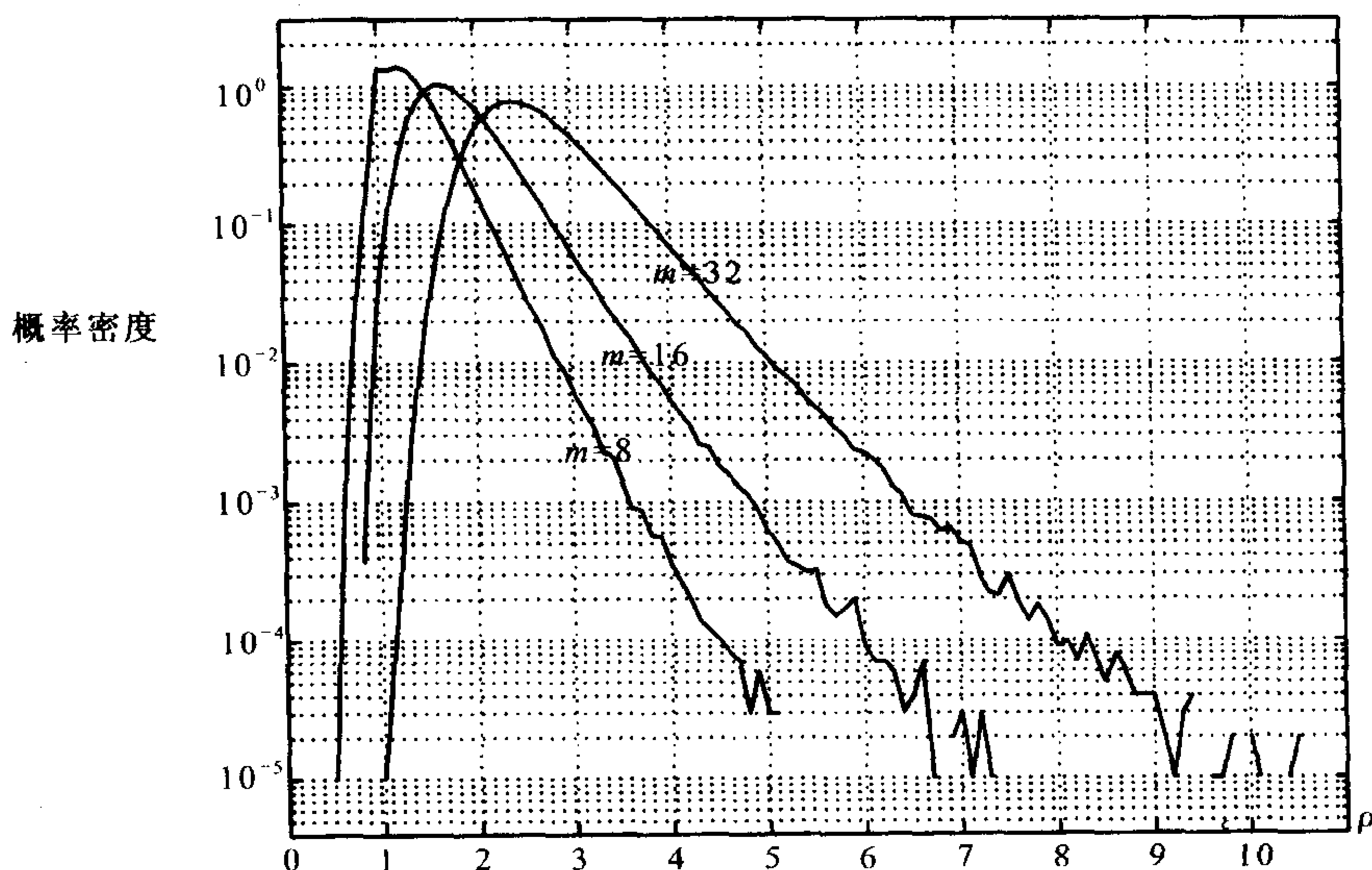


图 22-2 对维数 $m=8,16,32$ 的随机矩阵增长因子的概率密度分布，对每种维数基于 100 万的样本大小. 密度呈现出对 ρ 指数减少. 每条曲线接近末端的振动是有限样本大小的人为造成的

对用其他概率分布定义的随机矩阵可得到类似的结果，例如在 $[-1,1]$ 中一致分布的元素（习题 22.3)，如果你随机地选取 10 亿个矩阵，你几乎找不到一个对高斯消元法不稳定的矩阵.

22.5 解 释

我们不试图给出为什么对高斯消元法不稳定的矩阵是如此之少的一个完满的解释. 当然这也是不可能的，因为事情还未完全弄清楚. 但是我们仍将给出一个解释的轮廓.

167 ~ 168

如果 $PA=LU$，则 $U=L^{-1}PA$. 由此得出，若高斯消元法应用到矩阵 A 时不稳定(这隐含 ρ 是大的)，则 L^{-1} 也一定是大的. 现在，如果这样的情况出现，随机三角形矩阵有一个大的逆，而且作为维数 m 的函数呈指数式增大（习题 12.3（d)). 特别地，对由部分选主元高斯消元法提交的随机三角形矩阵是这样的，这个三角形矩阵对角线元素为 1 且对角线以下元素的绝对值小于等于 1.

然而，当高斯消元法应到随机矩阵 A 时，则得到的因子 L 是任意的但非随机的. 出现在 L 元素的符号之间的相关性反映了这些矩阵超常地良态. L^{-1} 的一个典型元素，远远不是指数式的大，而是它通常绝对值小于 1. 基于单个（但典型的）维数为 $m=128$ 的矩阵，图 22-3 给出了这种现象的迹象.

因此我们产生了这样的问题：为什么由高斯消元法提交的矩阵 L 几乎都没有大

的逆？

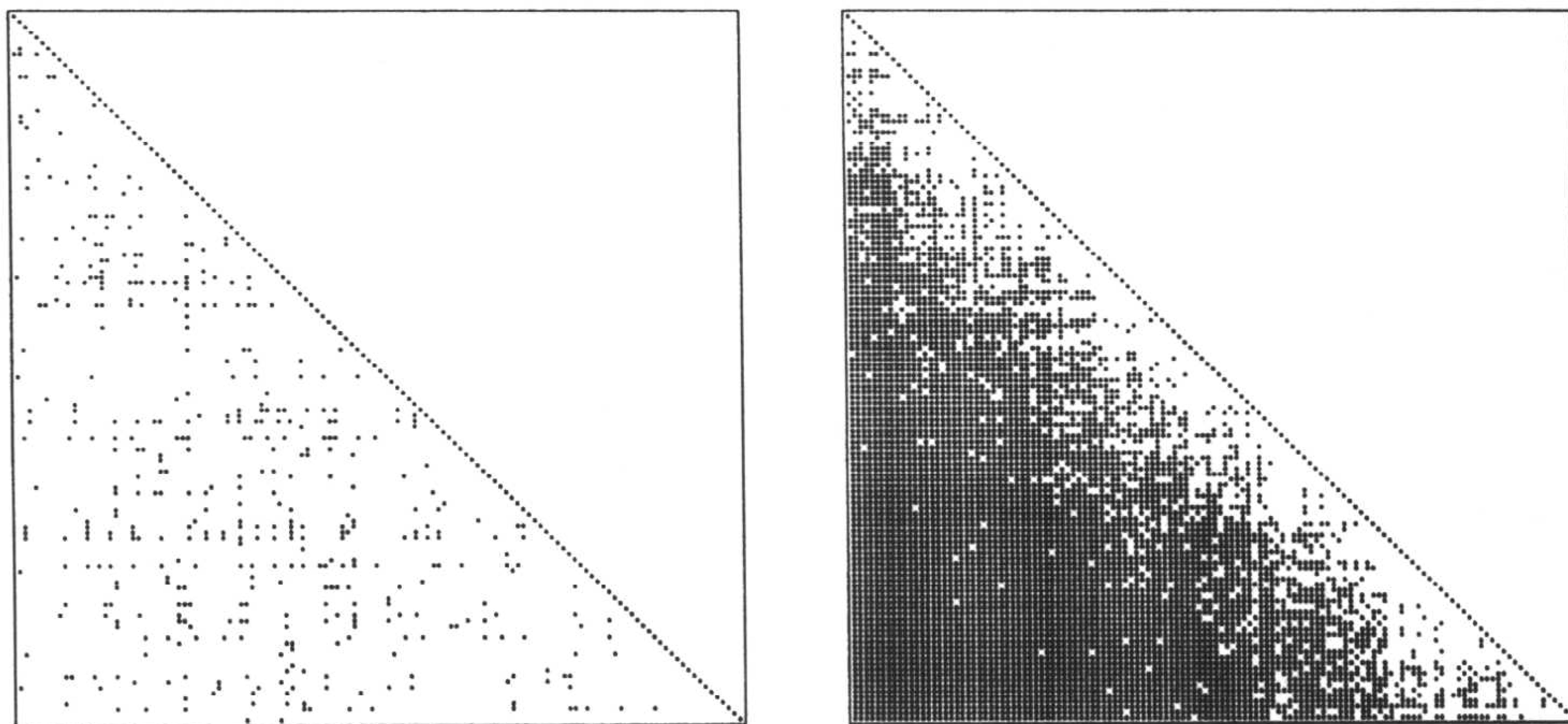


图 22-3 令 A 为一个随机的 128×128 矩阵，有因子分解 $PA=LU$. 图的左边显示了 L^{-1}：点表示绝对值大于等于 1 的元素. 图的右边是 $\tilde{L}^{-1}$类似的图形，其中 $\tilde{L}$ 和 L 是同样的，只是其对角线以下元素的符号被随机化了. 高斯消元法势必产生超常良态的矩阵 L

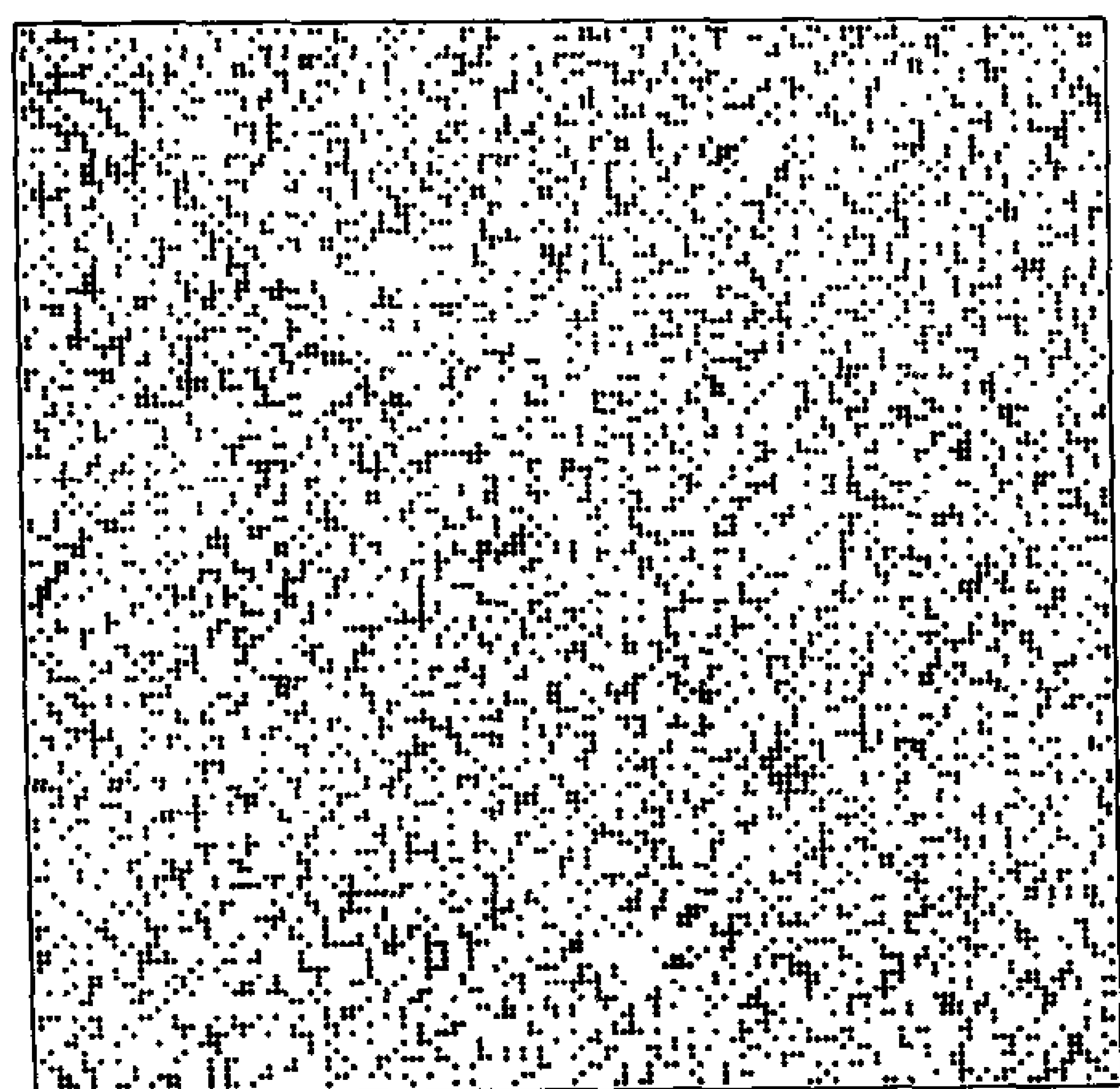
答案在对列空间的研究之中. 因为 U 是上三角的且 $PA=LU$，PA 和 L 的列空间是相同的. 由此意味着 PA 的第一列与 L 的第一列张成同样的空间，PA 的头两列与 L 的头两列张成同样的空间，如此等等. 如果 A 是随机的，则它的列空间被随机地取定，从而 $P^{-1}L$ 的列空间也一定如此. 然而，这个条件与 L^{-1} 是大的并不相容. 可以证明，若 L^{-1} 大，则 L 的列空间，或是任意置换 $P^{-1}L$ 的列空间，必定以一种非常远离随机性的方式变形.

图 22-4 证明了这点. 此图显示了与图 22-3 同样的两个矩阵的逐次列空间中其优点何在. 做到这点的工具是由 MATLAB 命令

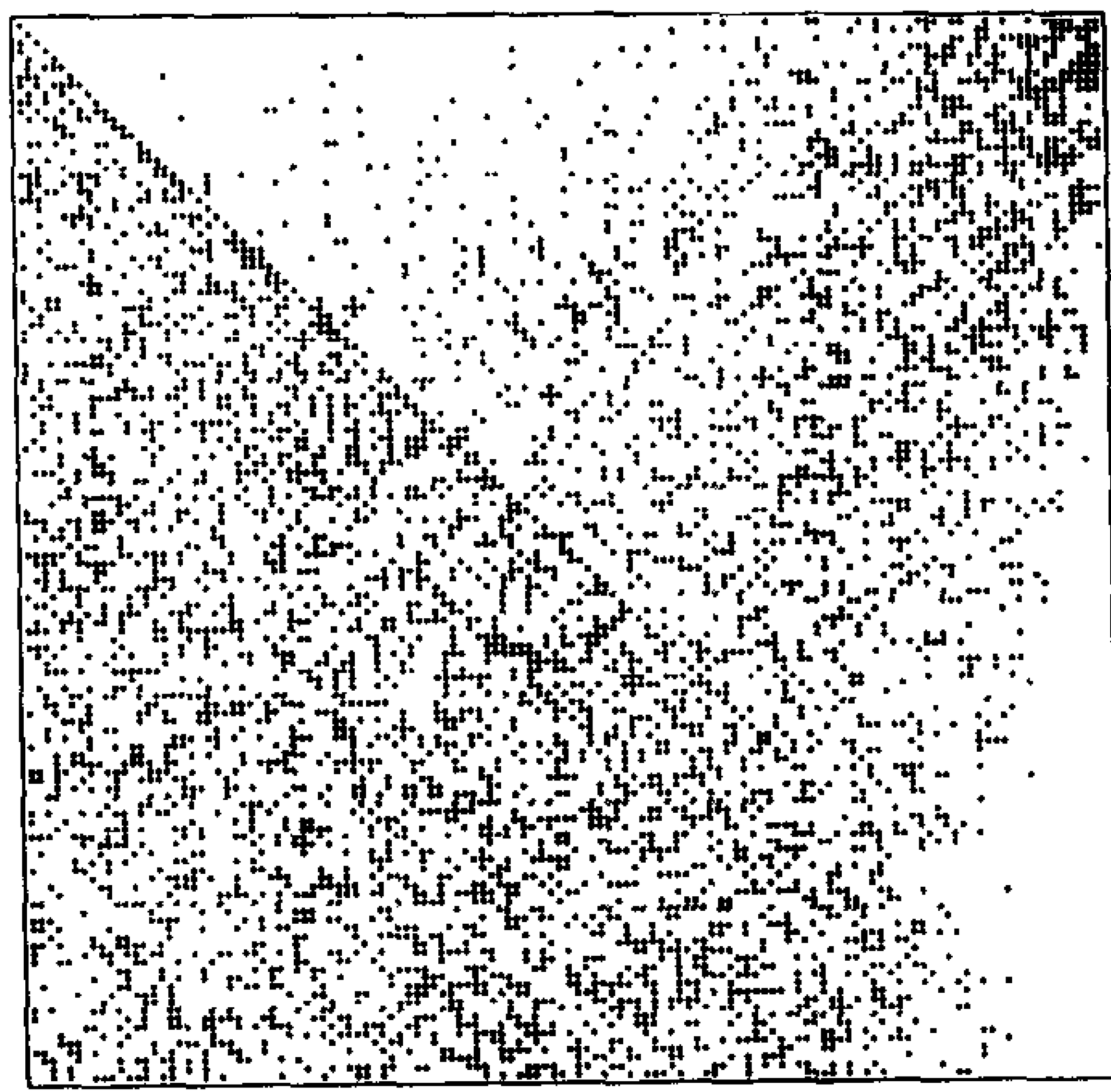
$$\texttt{[Q,R]=qr(A),}\quad \texttt{spy(abs(Q)>1/sqrt(m)).} \tag{22.6}$$

定义的一个 Q 描述（Q portrait）. 这些命令首先计算矩阵 A 的 QR 因子分解，然后在 Q 对应于元素大于标准差 $m^{-1/2}$的位置画一个点. 这个图说明，对随机的 A，甚至是行交换之后的形式 PA，其列空间的方向是接近随机的，而对给出大增长因子的矩阵 A，它远非随机. 可能量化这个描述，可以证明，在随机矩阵中大于 $m^{1/2}$阶的增长因子越来越稀少，其意义是对任意的 $\alpha>1/2$ 和 $M>0$，事件 $\rho>m^{\alpha}$ 的概率对所有充分大的 m 是小于 m^{-M}的. 然而，作为这种写法的一个定理仍然没有被证明出来.

169



随机的 A



随机的 $\tilde{L}$

图 22-4　同样两个矩阵的 Q 描述（22.6）．图的左边，随机矩阵 A 置换后成为 PA，或等价地，因子 L．图的右边，随机符号的矩阵 $\tilde{L}$．$\tilde{L}$ 的列空间以一种指数的方式变形，并不像在典型的随机矩阵类型中所产生的情况

我们来概括部分选主元高斯消元法的稳定性．这个算法对某些矩阵 A 是高度不稳定的，然而，对不稳定性的出现，A 的列空间一定以一种特别的方式变形，那至少在一类随机矩阵中越来越稀有．数十年的计算实验指出，列空间以这种形式的变形在应用上是非常少见的．

170

习　题

22.1　证明对部分选主元的高斯消元法用于任何矩阵 $A\in\mathbb{C}^{m\times m}$，增长因子（22.2）满足 $\rho\leqslant 2^{m-1}$．

22.2　对 A 有（22.4）形式的 60×60 方程组 $Ax=b$，用部分选主元的高斯消元法作求解实验．你是否注意到，因为增长因子为 2^{60} 阶而使得结果是无用的？在你最初的尝试中可能没有注意到这一点，因为 A 的整数元素可以防止任何舍入误差出现．如果这样，找一个方法稍微修改你的问题，使得增长因子是同样的或是相近的，且灾难性的舍入误差确确实实地发生了．

22.3　重新作出本讲的图，如果不能做到完全详细也要求近似的，但是要基于元素在 $[-1,1]$ 中均匀分布而不是正态分布的随机矩阵．你看到有任何本质的不同吗？

22.4　(a) 假设 $PA=LU$（部分选主元的 LU 因子分解）及 $A=QR$（QR 因子分解）．描述 L^{-1} 的最后一行和 Q 的最后一列之间的关系．

(b) 证明如果 A 在有独立的，正态分布的元素的意义下是随机的，则它的列空间随机地取定，特别是，这样 Q 的最后一列是一个随机单位向量．

(c) 结合 (a) 和 (b) 的结果，作出在高斯消元法应用到随机矩阵 A 中，关于 L^{-1} 的最后一列的一个陈述．

171

第 23 讲　楚列斯基因子分解

埃米尔特正定矩阵分解为三角因子的速度是一般矩阵的两倍. 对应的标准算法楚列斯基因子分解，是高斯消元法的一个变形，它在矩阵左右两边同时进行运算，保持并更好地利用了对称性.

23.1　埃米尔特正定矩阵

如果实矩阵 $A \in \mathbb{R}^{m\times m}$ 的对角线上和对角线下有相同的元素，即 $a_{ij}=a_{ji}$ 对所有 i，j 成立，那么 A 是对称的(symmetric)，因而有 $A=A^{\mathrm{T}}$. 这样的矩阵对所有向量 x，$y\in\mathbb{R}^m$ 满足 $x^{\mathrm{T}}Ay=y^{\mathrm{T}}Ax$.

对于复矩阵 $A\in\mathbb{C}^{m\times m}$，类似的性质是 A 为埃米尔特矩阵. 埃米尔特矩阵对角线下的元素是对角线上元素的复共轭，即 $a_{ij}=\overline{a_{ji}}$，因而有 $A=A^*$. （这些定义已经在第 2 讲中出现.）注意这意味着埃米尔特矩阵的对角元素一定是实的.

埃米尔特矩阵对所有 x，$y\in\mathbb{C}^m$ 满足 $x^*Ay=\overline{y^*Ax}$. 这意味着对于任意的 $x\in\mathbb{C}^m$，x^*Ax 是实的. 如果加上条件 $x^*Ax>0$ 对所有 $x\neq\mathbf{0}$ 成立，则称 A 为埃米尔特正定（有时只称正定）. 由于基本的物理学规律，很多在物理系统中产生的矩阵是埃米尔特正定的.

如果 A 是一个 $m\times m$ 埃米尔特正定矩阵，X 是一个 $m\times n$ 满秩矩阵，其中 $m\geqslant n$，则矩阵 X^*AX 也是埃米尔特正定的. 它是埃米尔特的是因为 $(X^*AX)^*=X^*A^*X=X^*AX$，它是正定的是因为对任意向量 $x\neq\mathbf{0}$，有 $Xx\neq\mathbf{0}$ 且因而有 $x^*(X^*AX)x=(Xx)^*A(Xx)>0$. 若 X 为一个每列有一个 1 而其他各处均为零的 $m\times n$ 矩阵，则可把 A 的任意 $n\times n$ 主子矩阵写成 X^*AX 的形式. 因此 A 的任意主子矩阵一定是正定的. 特别是，A 的每个对角元素都是一个正实数.　172

埃米尔特正定矩阵的特征根都是正实数. 如果 $Ax=\lambda x$ 对 $x\neq\mathbf{0}$ 成立，有 $x^*Ax=\lambda x^*x>0$，因而 $\lambda>0$. 反之，可以证明如果一个埃米尔特矩阵所有特征根都是正的，则它是正定的.

埃米尔特矩阵对应不同特征根的特征向量是正交的. [如下一讲所讨论的一样，埃米尔特矩阵是正规的(normal).] 设 $Ax_1=\lambda_1x_1$ 且 $Ax_2=\lambda_2x_2$，其中 $\lambda_1\neq\lambda_2$，则有

$$\lambda_2x_1^*x_2=x_1^*Ax_2=\overline{x_2^*Ax_1}=\overline{\lambda_1x_2^*x_1}=\lambda_1x_1^*x_2,$$

所以 $(\lambda_1-\lambda_2)\,x_1^*x_2=0$. 因为 $\lambda_1\neq\lambda_2$，所以有 $x_1^*x_2=0$.

23.2　对称高斯消元法

现在转到分解一个埃米尔特正定矩阵为三角形因子的问题. 一开始，我们考虑，如果把高斯消元法的一个单一步骤应用到左上角位置的元素为 1 的埃米尔特矩阵 A 将会怎样：

$$A=\begin{bmatrix}1 & w^* \\ w & K\end{bmatrix}=\begin{bmatrix}1 & 0 \\ w & I\end{bmatrix}\begin{bmatrix}1 & w^* \\ 0 & K-ww^*\end{bmatrix}.$$

如在第 20 讲所述，通过左乘一个初等下三角形运算使矩阵的第一列为零，这个运算使得第一行下面的各行减去第一行的倍数.

高斯消元法现在将由在第二列引入零的方式来继续约化为三角形形式. 然而为了保持对称性，楚列斯基因子分解首先在第一行引入零，对应于刚才在第一列引入零. 我们可以用使第一列后面各列减去第一列的倍数的右边一个上三角形运算来做到这一点：

$$\begin{bmatrix}1 & w^* \\ 0 & K-ww^*\end{bmatrix}=\begin{bmatrix}1 & 0 \\ 0 & K-ww^*\end{bmatrix}\begin{bmatrix}1 & w^* \\ 0 & I\end{bmatrix}.$$

注意，这个上三角形运算实际上是我们用于在第一列引入零的下三角形运算的一个伴随.

组合上述运算，我们发现矩阵 A 已经因子分解为 3 项：

173

$$A=\begin{bmatrix}1 & w^* \\ w & K\end{bmatrix}=\begin{bmatrix}1 & 0 \\ w & I\end{bmatrix}\begin{bmatrix}1 & 0 \\ 0 & K-ww^*\end{bmatrix}\begin{bmatrix}1 & w^* \\ 0 & I\end{bmatrix}. \tag{23.1}$$

楚列斯基因子分解的思想是继续这个过程，对称地在 A 的一列和一行引入零，直到它化为单位矩阵.

23.3　楚列斯基因子分解

为了将对称三角形约化在一般情形下进行，需要一个对任意 $a_{11}>0$ 的情况都起作用的因子分解，不仅只限在 $a_{11}=1$ 的情况. 其实用一个 $\sqrt{a_{11}}$ 的因子来调整 R_1 的某些元素就完成了 (23.1) 的推广. 令 $\alpha=\sqrt{a_{11}}$，看到

$$A=\begin{bmatrix}a_{11} & w^* \\ w & K\end{bmatrix}$$

$$=\begin{bmatrix}\alpha & 0 \\ w/\alpha & I\end{bmatrix}\begin{bmatrix}1 & 0 \\ 0 & K-ww^*/a_{11}\end{bmatrix}\begin{bmatrix}\alpha & w^*/\alpha \\ 0 & I\end{bmatrix}=R_1^*A_1R_1.$$

这是一个在楚列斯基因子分解中重复使用的基本步骤. 如果子矩阵 $\boldsymbol{K}-\boldsymbol{ww}^*/a_{11}$ 的左上角元素是正的，则同样的公式可以用于它的因子分解. 这样我们有 $\boldsymbol{A}_1=\boldsymbol{R}_2^*\boldsymbol{A}_2\boldsymbol{R}_2$，因而 $\boldsymbol{A}=\boldsymbol{R}_1^*\boldsymbol{R}_2^*\boldsymbol{A}_2\boldsymbol{R}_2\boldsymbol{R}_1$. 这个过程一直继续直到右下角，最终给出因子分解

$$\boldsymbol{A}=\underbrace{\boldsymbol{R}_1^*\boldsymbol{R}_2^*\cdots\boldsymbol{R}_m^*}_{\boldsymbol{R}^*}\underbrace{\boldsymbol{R}_m\cdots\boldsymbol{R}_2\boldsymbol{R}_1}_{\boldsymbol{R}}. \tag{23.2}$$

此方程形式为

$$\boldsymbol{A}=\boldsymbol{R}^*\boldsymbol{R},\qquad r_{jj}>0, \tag{23.3}$$

其中 $\boldsymbol{R}$ 为上三角形矩阵. 埃米尔特正定矩阵的这种约化称为**楚列斯基因子分解**(Cholesky factorization).

以上的描述留下了一则悬念，怎样知道子矩阵 $\boldsymbol{K}-\boldsymbol{ww}^*/a_{11}$ 的左上角元素是正的? 答案是因为 $\boldsymbol{K}-\boldsymbol{ww}^*/a_{11}$ 是正定矩阵 $\boldsymbol{R}_1^{-*}\boldsymbol{A}\boldsymbol{R}_1^{-1}$ 的 $(m-1)\times(m-1)$ 右下主子矩阵，所以它是正定的，所以其左上角元素为正. 用归纳法，同样的理由表明，出现在因子分解过程中的所有子矩阵 $\boldsymbol{A}_j$ 都是正定的，因此过程不会中断. 我们可以将这个结论正式写成为下面的定理.

定理 23.1 *每个埃米尔特正定矩阵 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$ 有惟一的楚列斯基因子分解* (23.3).

证明 存在性刚才已经讨论了. 因为算法不会中断，所以因子分解存在. 事实上，算法也建立了惟一性. 在每步 (23.2)，值 $\alpha=\sqrt{a_{11}}$ 由 $\boldsymbol{R}^*\boldsymbol{R}$ 因子分解的形式确定，一旦 α 被确定，$\boldsymbol{R}_1^*$ 的第一行也被确定. 因为类似的量在约化的每一步都被确定，所以整个因子分解是惟一的. □

174

23.4 算 法

当实现楚列斯基因子分解时，运算中的矩阵只有一半需要显式地表示. 这种简化减少了一半算法. 算法的一个正式陈述（只是若干种可能的一种）在下面给出. 输入矩阵 $\boldsymbol{A}$ 表示被因子分解的 $m\times m$ 埃米尔特正定矩阵对角线以上的一半，（在实际的软件中，可以用一种压缩存储格式来避免浪费一个正方形数组的一半元素.）输出矩阵 $\boldsymbol{R}$ 表示 $\boldsymbol{A}=\boldsymbol{R}^*\boldsymbol{R}$ 的上三角形因子. 每次外迭代对应单个基本的因子分解：子矩阵 $\boldsymbol{R}_{k:m,k:m}^*$ 的上三角部分表示，在第 k 步被分解的埃米尔特矩阵的对角线的以上部分.

算法 23.1 楚列斯基因子分解

$R=A$
for $k=1$ **to** m
  for $j=k+1$ **to** m
    $R_{j,j:m}=R_{j,j:m}-R_{k,j:m}\overline{R}_{kj}/R_{kk}$
  $R_{k,k:m}=R_{k,k:m}/\sqrt{R_{kk}}$

23.5 运算计数

在楚列斯基因子分解中，算法所起的作用主要在内循环．单独执行

$$R_{j,j:m}=R_{j,j:m}-R_{k,j:m}\overline{R}_{kj}/R_{kk}$$

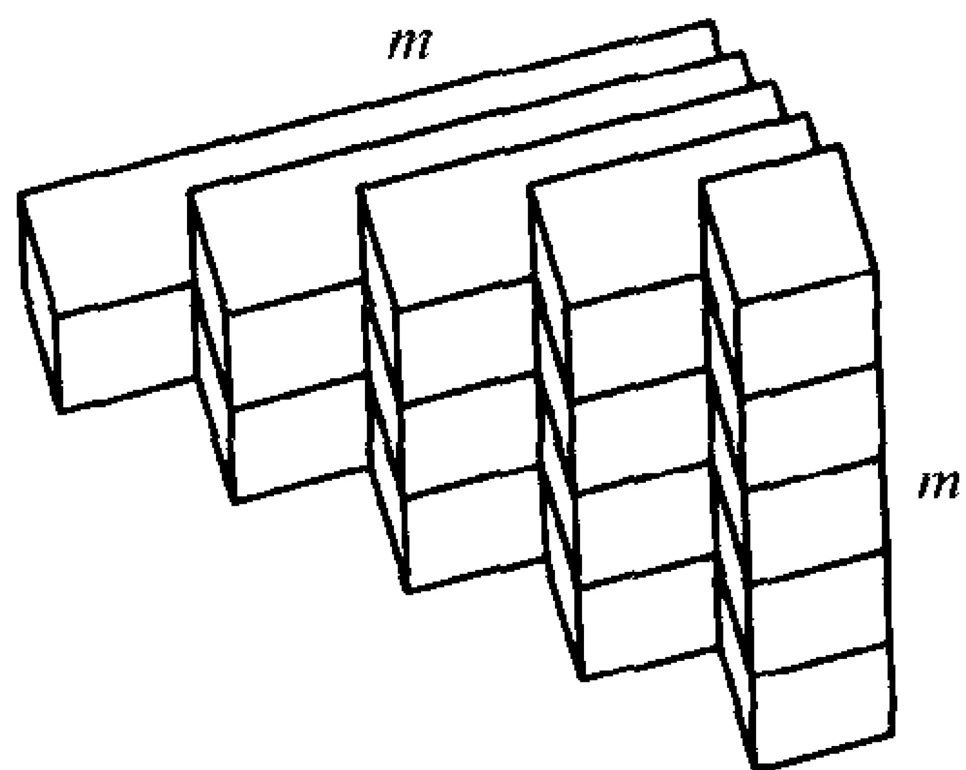
这一行需要一次除法，$m-j+1$ 次乘法和 $m-j+1$ 次减法，总共是 $\sim 2(m-j)$ 次 flop. 这个计算对由 $k+1$ 到 m 的每个 j 重复一次，循环对由 1 到 m 的每个 k 重复一次．直接计算出其和：

$$\sum_{k=1}^{m}\sum_{j=k+1}^{m}2(m-j)\sim 2\sum_{k=1}^{m}\sum_{j=1}^{k}j\sim\sum_{k=1}^{m}k^2\sim\frac{1}{3}m^3 \text{ 次 flop.}$$

因此，楚列斯基因子分解只包含高斯消元法众多运算的一半，后者对分解同样的矩阵需要 $\sim\frac{2}{3}m^2$ 次 flop.

175

像通常那样，运算计数也可以由图形确定．对每个 k，在一个三角形层的每个位置产生 2 次浮点运算（一次乘法和一次减法）．完整的算法对应 m 层的堆砌：



当 $m\to\infty$ 时，几何体收敛到体积为$\frac{1}{6}m^3$ 的四面体．因为每个单位立方体对应 2 次浮点运算，我们再次得到

$$\text{楚列斯基因子分解的工作量：}\sim\frac{1}{3}m^3 \text{ 次 flop.}\tag{23.4}$$

23.6 稳　定　性

对于楚列斯基因子分解，高斯消元法稳定性分析的所有难以捉摸的问题都消失了．这个算法总是稳定的．直观上看，理由是，因子 $\boldsymbol{R}$ 永远不会增长变大．例如，对 2-范数，有 $\|\boldsymbol{R}\|=\|\boldsymbol{R}^*\|=\|\boldsymbol{A}\|^{1/2}$（证明：SVD），对其他的 p-范数($1\leqslant p\leqslant\infty$)，$\|\boldsymbol{R}\|$与$\|\boldsymbol{A}\|^{1/2}$不能有大于$\sqrt{m}$的差别因子．因此，比 $\boldsymbol{A}$ 的元素大很多的数永远不会产生．

注意，楚列斯基因子分解的稳定性不需要任何选主元就可达到. 直观上看，它与这样的事实是相关的，即一个埃米尔特正定矩阵大部分的权重是在其对角线上. 例如，不难证明最大元素一定在对角线上，并可将这个性质推广到归纳过程（23.2）所构造的正定子矩阵上.

楚列斯基因子分解的稳定性分析导致了如下的向后稳定性结论.

定理 23.2　令 $A\in\mathbb{C}^{m\times m}$ 为埃米尔特正定，且 A 的楚列斯基因子分解由算法 23.1 在一个满足（13.5）和（13.7）的计算机上计算. 对于充分小的 $\epsilon_{机器}$，保证过程不被间断（也即没有零或负的角元素 r_{kk} 产生），产生一个计算出的因子 $\tilde{R}$，对某个 $\delta A\in\mathbb{C}^{m\times m}$ 满足

$$\tilde{R}^*\tilde{R}=A+\delta A,\qquad \frac{\|\delta A\|}{\|A\|}=O(\epsilon_{机器}) \tag{23.5}$$

176

像数值线性代数很多的算法一样，对这个算法，如果我们企图作出一个向前误差分析而不是一个向后的分析，情况会坏得多. 如果 A 是病态的，$\tilde{R}$ 一般不会接近 R. 我们可以说最好的情况是 $\|\tilde{R}-R\|/\|R\|=O(\kappa(A)\epsilon_{机器})$，（换句话说，一般楚列斯基因子分解是一个病态问题.）只有乘积 $\tilde{R}^*\tilde{R}$ 才满足相当好的误差界（23.5）. 因此在 $\tilde{R}$ 中由舍入所引起的误差是大的，但它是“魔鬼相关”的，正如在第 16 讲中对 QR 因子分解所看到的那样.

23.7　$Ax=b$ 的解

如果 A 是埃米尔特正定的，解方程组 $Ax=b$ 的标准方法是用楚列斯基因子分解. 算法 23.1 把方程组化为 $R^*Rx=b$，因而我们逐次解两个三角形方程组：首先对未知的 y 解 $R^*y=b$，然后对未知的 x 解 $Rx=y$. 每次解三角形方程组只需 $\sim m^2$ 次 flop，所以全部工作再次是 $\sim\frac{1}{3}m^3$ 次 flop.

由类似于第 16 讲中所述的理由，可以表明这个过程是向后稳定的.

定理 23.3　埃米尔特正定方程组 $Ax=b$ 用楚列斯基因子分解（算法 23.1）的解是向后稳定的，产生一个计算出的解 $\tilde{x}$，对某个 $\Delta A\in\mathbb{C}^{m\times m}$ 满足

$$(A+\Delta A)\tilde{x}=b,\qquad \frac{\|\Delta A\|}{\|A\|}=O(\epsilon_{机器}). \tag{23.6}$$

习　题

23.1　令 A 为非奇异正方形矩阵，且令 $A=QR$ 和 $A^*A=U^*U$ 分别是 QR 和楚列斯基因子分

解，用通常的正规化使 $r_{jj}, u_{jj} > 0$. $\boldsymbol{R} = \boldsymbol{U}$ 是否正确？

23.2　仿造定理 16.2 的证明方法，由定理 23.1 和定理 17.1 推导定理 23.3.

23.3　"\" 的反向软件工程. 下面的 MATLAB 对话记录了一个 1991 年制造的工作站上各种计算运行时间的系列测试. 对每一部分，尝试解释：(i)为什么要做这个实验？(ii) 为什么会出现这个结果？你的答案可以参考课文中关于运算计数的公式. MATLAB 的查询 help chol 和 help slash 在你的探究工作中会有所帮助.

177

(a)
```
m = 200; Z = randn(m,m);
A = Z'*Z; b = randn(m,1);
tic; x = A\b; toc;
        elapsed_time = 1.0368
```

(b)
```
tic; x = A\b; toc;
        elapsed_time = 1.0303
```

(c)
```
A2 = A; A2(m,1) = A2(m,1)/2;
tic; x = A2\b; toc;
        elapsed_time = 2.0361
```

(d)
```
I = eye(m,m); emin = min(eig(A));
A3 = A - .9*emin*I;
tic; x = A3\b; toc;
        elapsed_time = 1.0362
```

(e)
```
A4 = A - 1.1*emin*I;
tic; x = A4\b; toc;
        elapsed_time = 2.9624
```

(f)
```
A5 = triu(A);
tic; x = A5\b; toc;
        elapsed_time = 0.1261
```

(g)
```
A6 = A5; A6(m,1) = A5(1,m);
tic; x = A6\b; toc;
        elapsed_time = 2.0012
```

178

第V部分

特　征　值

第 24 讲　特征值问题

由于求解特征值的最优算法很有效，但不是一般人能立刻想到的，因此特征值问题在科学计算中引起了人们的特别兴趣. 在此，我们复习一下特征值和特征向量的数学知识. 有关算法将在下面几讲中讨论.

24.1　特征值和特征向量

设 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$ 为方阵，如果对于 $\lambda\in\mathbb{C}$，存在一个非零向量 $\boldsymbol{x}\in\mathbb{C}^m$ 使得

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}. \tag{24.1}$$

则称 λ 为 $\boldsymbol{A}$ 的一个特征值（eigenvalue），而 $\boldsymbol{x}$ 为相应的特征向量（eigenvector）. 这里的思路是，矩阵 $\boldsymbol{A}$ 作用在 $\mathbb{C}^m$ 的子空间 $\boldsymbol{S}$ 上，有时可以模拟标量乘法. 若是这样，这个特殊的子空间 $\boldsymbol{S}$ 称为特征空间（eigenspace），且任意非零向量 $\boldsymbol{x}\in\boldsymbol{S}$ 为特征向量.

矩阵 $\boldsymbol{A}$ 的全部特征值组成的集合称为 $\boldsymbol{A}$ 的谱（spectrum），它是 $\mathbb{C}$ 的子集，用 $\Lambda(\boldsymbol{A})$ 表示.

与前几讲讨论过的包含方形或长方形方程组的问题相比，特征值问题有一个很不同的特征，对于方程组，$\boldsymbol{A}$ 的定义域和值域可能是不同的空间. 比如，在例 1.1 中，$\boldsymbol{A}$ 把多项式系数的 n 维向量映射成样本多项式值的 m 维向量. 讨论这样一个矩阵 $\boldsymbol{A}$ 的特征值是毫无意义的. 仅当矩阵的值域空间和定义域空间相同时，特征值问题才有意义. 这反映了一个事实，在应用中，特征值一般是用于一个矩阵反复地复合的情况，如应用于显示函数幂 $\boldsymbol{A}^k$，或隐式函数 $e^{t\boldsymbol{A}}$.

一般地说，特征值和特征向量可用于两个方面，其一为算法的，其二为物理的. 在算法方面，特征值分析可以用约化一个耦合方程组为一族标量方程的办法来简化某些问题的解法；在物理方面，用特征值分析能深刻理解由线性方程组决定的演化系统的性质. 后一类最熟悉的例子是对共振（resonance）（例如，当打击、弹和拉乐器时，乐器产生的共振）和稳定性（stability）（例如，在小扰动条件下流体流动的稳定性）的研究. 在上述情况中，对于分析长时间 t 的性质，特征值是特别有用的，见习题 24.3.

24.2　特征值分解

方阵 $\boldsymbol{A}$ 的特征值分解（eigenvalue decomposition），已在（5.1）中讨论过，它是

一种因子分解

$$A = X\Lambda X^{-1}. \tag{24.2}$$

(正如下面所讨论，这样的因子分解并非总存在.)其中 X 是非奇异的矩阵，Λ 为对角矩阵.

这一定义可以重新写为

$$AX = X\Lambda, \tag{24.3}$$

即

$$\left[\begin{array}{c} \\ A \\ \\ \end{array}\right]\left[\begin{array}{c|c|c|c} & & & \\ x_1 & x_2 & \cdots & x_m \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ x_1 & x_2 & \cdots & x_m \\ & & & \end{array}\right]\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix}.$$

显然，如果 x_j 是 X 的第 j 列，λ_j 是 Λ 对角线上第 j 个元素，那么 $Ax_j=\lambda_j x_j$. 于是，X 的第 j 列是 A 的特征向量，Λ 对角线上第 j 个元素是对应的特征值.

特征值分解表示的是基到“特征向量坐标”的变换. 如果 $Ax=b$ 和 $A=X\Lambda X^{-1}$，那么有

$$(X^{-1}b)=\Lambda(X^{-1}x). \tag{24.4}$$

于是，要计算 Ax，可以把 x 在以 X 的列向量组成的基中展开，应用 Λ，可把这结果解释为由 X 的列向量的线性组合的系数组成的向量.

182

24.3 几何重数

如前所述，与单个特征值相对应的特征向量的集合加上零向量构成了 $\mathbb{C}^m$ 的子空间，称为特征空间. 如果 λ 为 A 的一个特征值，那么用 E_λ 来表示相应的特征空间. 特征空间 E_λ 是 A 的一个不变子空间 (invariant subspace)，即 $AE_\lambda\subseteq E_\lambda$.

E_λ 的维数可以解释为与同一特征值 λ 相对应的线性无关特征向量的最大个数，这个数称为 λ 的几何重数 (geometric multiplicity). 由于 $A-\lambda I$ 的零空间为 E_λ，所以 λ 的几何重数可以称为 $A-\lambda I$ 的零空间的维数.

24.4 特征多项式

$A\in\mathbb{C}^{m\times m}$ 的特征多项式 (characteristic polynomial) 是 m 次多项式，用 p_A 或简单地用 p 来表示，定义为

$$p_A(z)=\det(zI-A). \tag{24.5}$$

由于对负号作了处理，所以 p 是首 1 多项式（monic）：次数为 m 的项系数为 1.

定理 24.1 λ 是 A 的特征值的充分必要条件为 $p_A(\lambda)=0$.

证明 可由特征值的定义得出：

$$\begin{aligned}\lambda \text{ 是特征值} &\iff \text{存在一个非零向量 } \boldsymbol{x} \text{ 使得 } \lambda\boldsymbol{x}-\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0}\\ &\iff \lambda\boldsymbol{I}-\boldsymbol{A} \text{ 是奇异的}\\ &\iff \det(\lambda\boldsymbol{I}-\boldsymbol{A})=0.\end{aligned}$$

□

定理 24.1 具有一个重要的推论. 即使矩阵是实的，它的一些特征值可以是复的. 在物理上，这是与实动力系统出现的或增或减振荡运动有关. 在算法上，意味着即使输入的特征值问题是实的，输出的值也可能是复的.

24.5 代数重数

根据代数学的基本定理，可以把 p_A 写成如下形式，对于某些 $\lambda_j\in\mathbb{C}$，有

$$p_A(z)=(z-\lambda_1)(z-\lambda_2)\cdots(z-\lambda_m) \tag{24.6}$$

利用定理 24.1 得出，每个 λ_j 是 A 的特征值，并且 A 的所有特征值均出现在这个列举式中. 一般地，一个特征值出现可以多于一次. 将 A 的特征值 λ 的代数重数（algebraic multiplicity）定义为 λ 作为 p_A 的根的重数. 如果特征值的代数重数为 1，那么这个特征值是单的（simple）. 183

特征多项式提供了对于一个矩阵的特征值进行计数的简单办法.

定理 24.2 如果 $A\in\mathbb{C}^{m\times m}$，那么在计代数重数下，$A$ 有 m 个特征值，特别地，如果 p_A 的根都是单根，那么 A 有 m 个不同的特征值.

特别应注意，每个矩阵至少有一个特征值.

一个特征值的代数重数均大于等于其几何重数，为证明此结论，需知道一些关于相似变换的知识.

24.6 相似变换

如果 $X\in\mathbb{C}^{m\times m}$ 是非奇异的，映射 $A\to X^{-1}AX$ 称为 A 的相似变换（similarity transformation）. 如果存在一个相似变换把矩阵 A 与矩阵 B 联系起来，换句话说，如果存在一个非奇异矩阵 $X\in\mathbb{C}^{m\times m}$ 使得 $B=X^{-1}AX$，那么称 A 和 B 是相似的（similar）. 如前面在对角化（24.2）的特殊情况下所讨论的，任何一个相似变换都是一个基运算变换.

相似矩阵 A 和 $X^{-1}AX$ 有许多相同的性质.

定理 24.3 如果 X 为非奇异的，那么 A 和 $X^{-1}AX$ 有相同的特征多项式、特征值、以及代数和几何重数.

证明 特征多项式的匹配关系式的证明可由直接计算得到：

$$\begin{aligned} p_{X^{-1}AX}(z) &= \det(z\boldsymbol{I}-\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}) = \det(\boldsymbol{X}^{-1}(z\boldsymbol{I}-\boldsymbol{A})\boldsymbol{X}) \\ &= \det(\boldsymbol{X}^{-1})\det(z\boldsymbol{I}-\boldsymbol{A})\det(\boldsymbol{X}) = \det(z\boldsymbol{I}-\boldsymbol{A}) = p_A(z). \end{aligned}$$

由特征多项式相同，特征值和代数重数相同立即可得. 最后，为证明几何重数的一致性. 可证明，如果 $\boldsymbol{E}_\lambda$ 是 $\boldsymbol{A}$ 的特征空间，那么 $\boldsymbol{X}^{-1}\boldsymbol{E}_\lambda$ 是 $\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$ 的特征空间，反之亦然. □

下面讨论几何重数和代数重数的关系.

184 **定理 24.4** 特征值 λ 的代数重数大于等于其几何重数.

证明 设 n 为矩阵 $\boldsymbol{A}$ 的特征值 λ 的几何重数. 构造 $m\times n$ 矩阵 $\hat{\boldsymbol{V}}$，其 n 列构成了特征空间 $\{\boldsymbol{x}:\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}\}$ 的一组标准正交基. 将 $\hat{\boldsymbol{V}}$ 扩充成方的酉矩阵 $\boldsymbol{V}$，可以得到 $\boldsymbol{V}^*\boldsymbol{A}\boldsymbol{V}$ 有如下形式

$$\boldsymbol{B} = \boldsymbol{V}^*\boldsymbol{A}\boldsymbol{V} = \begin{bmatrix} \lambda\boldsymbol{I} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{D} \end{bmatrix}, \tag{24.7}$$

其中 $\boldsymbol{I}$ 为 $n\times n$ 的单位矩阵，$\boldsymbol{C}$ 是 $n\times(m-n)$ 的矩阵，$\boldsymbol{D}$ 为 $(m-n)\times(m-n)$ 的矩阵. 由行列式的定义，有 $\det(z\boldsymbol{I}-\boldsymbol{B})=\det(z\boldsymbol{I}-\lambda\boldsymbol{I})\det(z\boldsymbol{I}-\boldsymbol{D})=(z-\lambda)^n\det(z\boldsymbol{I}-\boldsymbol{D})$. 因此，作为 $\boldsymbol{B}$ 的特征值 λ 的代数重数至少为 n. 由于相似变换保持重数不变，所以此结论对 $\boldsymbol{A}$ 也成立. □

24.7 亏损特征值和亏损矩阵

虽然泛矩阵有相等的代数重数和几何重数（即都为 1），但决不是对每个矩阵都这样.

例 24.1 考虑矩阵

$$\boldsymbol{A} = \begin{bmatrix} 2 & & \\ & 2 & \\ & & 2 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 2 & 1 & \\ & 2 & 1 \\ & & 2 \end{bmatrix}.$$

$\boldsymbol{A}$ 和 $\boldsymbol{B}$ 有相同的特征多项式 $(z-2)^3$，所以有代数重数为 3 的单特征值 $\lambda=2$. 对于 $\boldsymbol{A}$，可以选择 3 个独立的特征向量，例如 $\boldsymbol{e}_1,\boldsymbol{e}_2$ 和 $\boldsymbol{e}_3$，所以几何重数也是 3. 另一方面，对于 $\boldsymbol{B}$，仅能找到一个独立的特征向量，（$\boldsymbol{e}_1$ 的数量倍），所以特征值的几何重数仅为 1. □

代数重数超过几何重数的特征值称为**亏损特征值**（defective eigenvalue），有一个或多个亏损特征值的矩阵称为**亏损矩阵**（defective matrix）.

任何对角矩阵都是非亏损的，对于这样的矩阵，特征值 λ 的代数重数和几何重

数均等于 λ 在对角线上出现的个数.

24.8 可对角化

非亏损矩阵类确切地说，就是具有特征值分解（24.2）的矩阵类.

定理 24.5 *$m \times m$ 矩阵 A 是非亏损的充分必要条件是矩阵有特征值分解 $A = X\Lambda X^{-1}$.* 185

证明 （$\Leftarrow$）给定 A 的特征值分解 $A = X\Lambda X^{-1}$，由定理 24.3 得出，Λ 相似于 A，并有相同的特征值和相同的重数. 因为 Λ 是对角矩阵，所以它是非亏损的，于是 A 也是非亏损的.

（$\Rightarrow$）由于属于不同特征值的特征向量是线性无关的，并且属于每个特征值的线性无关的向量数等于特征值的重数，因此非亏损矩阵必有 m 个线性无关的特征向量. 如果将这 m 个线性无关的特征向量组成矩阵 X 的列，那么 X 是非奇异的，并且有 $A = X\Lambda X^{-1}$. □

根据上述结论，非亏损的另一个术语为可对角化（diagonalizable）.

可对角化矩阵 A 是否在某种意义下具有其对角等价矩阵 Λ 的特征？这个回答取决于人们度量其性质的哪个方面，和特征向量构成的矩阵 X 的条件数. 如果 X 是高度病态的，那么在 A 变换到 Λ 时会失去大量信息. 见第 34 讲的"警示：非正规性".

24.9 行列式和迹

矩阵 $A \in \mathbb{C}^{m\times m}$ 的迹（trace）是其对角线元素的和：$\mathrm{tr}(A) = \sum_{j=1}^{m} a_{jj}$，迹和行列式与特征值有如下关系：

定理 24.6 *考虑代数重数情况下，行列式 $\det(A)$ 与迹 $\mathrm{tr}(A)$ 分别等于 A 的特征值乘积与和：*

$$\det(A) = \prod_{j=1}^{m} \lambda_j, \qquad \mathrm{tr}(A) = \sum_{j=1}^{m} \lambda_j. \tag{24.8}$$

证明 从（24.5）和（24.6），计算

$$\det(A) = (-1)^m \det(-A) = (-1)^m p_A(0) = \prod_{j=1}^{m} \lambda_j.$$

这就建立了第一个公式. 至于第二个公式，从（24.5）可以得出，p_A 的含 z^{m-1} 项的系数是 A 的对角线元素之和的相反数，即 $-\mathrm{tr}(A)$. 另一方面，从（24.6）得，这系数也等于 $-\sum_{j=1}^{m} \lambda_j$，于是有 $\mathrm{tr}(A) = \sum_{j=1}^{m} \lambda_j$. □

24.10　酉对角化

有时，不仅要求 $m \times m$ 矩阵 A 有 m 个线性无关的特征向量，而且还要求所求的 186 向量为正交的. 若上述条件满足，那么 A 是可酉对角化的（unitarily diagonalizable），即，存在一个酉矩阵 Q，使得

$$A = Q\Lambda Q^{*}. \tag{24.9}$$

若不考虑 Λ 的元素的符号问题（可能为复数），这个因子分解既是特征值分解又是奇异值分解.

曾见过一类矩阵是可酉对角化的：埃米尔特矩阵. 下面结果可从以后将讨论的定理 24.9 得到.

定理 24.7　*埃米尔特矩阵是可酉对角化矩阵，且其特征值是实的.*

埃米尔特矩阵不是惟一可酉对角化的矩阵，其他的例子包括反埃米尔特矩阵、酉矩阵、循环矩阵以及任一这样的矩阵加上单位矩阵的倍数构成的矩阵. 一般地，可以酉对角化的矩阵类均具有良好的特征，如果 $A^{*}A = AA^{*}$，那么定义矩阵 A 为正规的（normal）. 从而得到如下著名的结果.

定理 24.8　*矩阵是可酉对角化的充分必要条件是它是正规矩阵.*

24.11　舒尔因子分解

最后讨论的矩阵因子分解实际上是数值分析中最有用的，其原因是，包括亏损矩阵在内的所有矩阵都可以用这个方法实现因子分解. 矩阵 A 的舒尔因子分解（Schur factorization）是定义如下

$$A = QTQ^{*}, \tag{24.10}$$

其中，Q 为酉矩阵，T 为上三角矩阵. 注意到，A 与 T 是相似的，所以 A 的特征值必定出现在 T 的对角线上.

定理 24.9　*每个方阵 A 都存在舒尔因子分解.*

证明　对矩阵 A 的维数 m 用归纳法来证明. $m=1$ 是平凡的，所以假定 $m \geqslant 2$. 设 x 为 A 的任一特征向量，与之相对应的特征值为 λ. 把 x 规范化，并令它为酉矩阵 U 的第一列，如（24.7）一样，容易验证积 $U^{*}AU$ 有如下形式

$$U^{*}AU = \begin{bmatrix} \lambda & B \\ 0 & C \end{bmatrix}.$$

用归纳假设，存在 C 的舒尔因子分解 VTV^{*}. 记

$$Q = U\begin{bmatrix} 1 & 0 \\ 0 & V \end{bmatrix}.$$

187

这是一个酉矩阵，并有

$$Q^* AQ = \begin{bmatrix} \lambda & BV \\ 0 & T \end{bmatrix}.$$

这就是所求的舒尔分解. □

24.12　显现特征值的因子分解

在前几页中，已经描述了显现特征值因子分解的 3 个例子，矩阵的因子分解约化为使特征值显式地表达的形式. 对此总结如下：

A 对角化 $A = X\Lambda X^{-1}$ 存在的充分必要条件为 A 为非亏损的.

A 酉对角化 $A = Q\Lambda Q^*$ 存在的充分必要条件为 A 为正规矩阵.

A 酉三角化（舒尔因子分解）$A = QTQ^*$ 总存在.

为计算特征值，将建立一种这样的因子分解. 由于舒尔因子分解将无限制地应用到所有矩阵上，所以一般将使用舒尔因子分解. 而且，由于包含了酉变换，所以算法在数值上是稳定的. 如果 A 是正规矩阵，那么舒尔型为对角形，特别地，如果 A 是埃米尔特矩阵，那么在整个计算中可利用其对称性的优点，并且把 A 约化为对角形式的矩阵所需的工作仅为一般矩阵 A 的一半或更少.

习　　题

24.1　对于下述每一个结论，试证明它是正确的，或举一个例子来说明它是错误的. 本题中，除非特别说明，$A \in \mathbb{C}^{m\times m}$，“ew” 表示特征值.（这源自德语 “Eigenwert”，特征向量的相应缩写是 “ev”，源自 “Eigenvektor”.）

(a) 如果 λ 是 A 的一个 ew，$\mu \in \mathbb{C}$，那么 $\lambda - \mu$ 是 $A - \mu I$ 的一个 ew.

(b) 如果 A 是实的，λ 为 A 的一个 ew，那么 $-\lambda$ 也是 A 的一个 ew.

(c) 如果 A 是实的，λ 为 A 的一个 ew，那么 $\bar{\lambda}$ 也是 A 的一个 ew.

(d) 如果 λ 是 A 的一个 ew，A 非奇异，那么 λ^{-1} 是 A^{-1} 的一个 ew.

(e) 如果 A 的所有 ew 为零，那么 $A = 0$.

(f) 如果 A 是埃米尔特矩阵，λ 是 A 的一个 ew，那么 $|\lambda|$ 是 A 的一个奇异值.

(g) 如果 A 是可对角化的，并且其所有 ew 都相等，那么 A 是对角矩阵.

188

24.2　下面是 Gerschgorin 定理，它对于任意 $m\times m$ 的矩阵 A，无论对称的或非对称的都成立. A 的每个特征值至少位于复平面上中心为 a_{ii}，半径为 $\sum_{j\neq 1} |a_{ij}|$ 的 m 个圆盘中的一个上，而且，如果 m 个圆盘中的 n 个圆盘形成一个连通区域，且与其他 $m-n$ 个圆盘不相连

接，则在这区域中恰有 A 的 n 个特征值.

(a) 证明 Gerschgorin 定理的第一部分.（提示：令 λ 为 A 的任一特征值，x 为对应的具有最大分量为 1 的特征向量.）

(b) 证明第二部分.（提示：把 A 变形为对角矩阵，并利用矩阵的特征值是其元素的连续函数这一事实.）

(c) 设

$$A=\begin{pmatrix}8 & 1 & 0\\ 1 & 4 & \epsilon\\ 0 & \epsilon & 1\end{pmatrix}, |\epsilon|<1,$$

用 Gerschgorin 定理估计其特征值.

(d) 找一种方法，建立关于 A 的最小特征值的更为严格的界 $|\lambda_3-1|\leq\epsilon^2$.（提示：考虑对角相似变换.）

24.3 设 A 是元素由标准正态分布得到的 10×10 的随机矩阵减去单位阵的 2 倍，编写一个程序，并绘出 $\|e^{tA}\|_2$ 关于 t，$0\leq t\leq 20$ 在对数尺度下的图形，并与直线 $e^{t\alpha(A)}$ 的结果作比较，其中 $\alpha(A)=\max_j \mathrm{Re}(\lambda_j)$ 是 A 的谱横坐标（spectral abscissa）. 对 10 个随机矩阵运行程序并对结果做评论. 矩阵的什么性质导致了曲线 $\|e^{tA}\|_2$ 当 $t\to\infty$ 时仍是振荡的.

24.4 对任意 $A\in\mathbb{C}^{m\times m}$ 和范数 $\|\cdot\|$，用定理 24.9 证明：

(a) $\lim_{n\to\infty}\|A^n\|=0 \iff \rho(A)<1$，其中 ρ 为谱半径（习题 3.2）.

189

(b) $\lim_{t\to\infty}\|e^{tA}\|=0 \iff \alpha(A)<0$，其中 α 为谱横坐标.

第 25 讲　特征值算法综述

这一讲及其后 5 讲中，将讨论计算特征值和特征向量的一些经典的“直接”算法，以及几个近代算法. 大部分算法都包含两个阶段：首先，把一个满秩矩阵经初步约化成特定形式；然后是为得到最后收敛的迭代过程. 这一讲简要地介绍这两个阶段的方法以及它们的优点.

25.1　显而易见的算法的缺点

虽然特征值和特征向量定义简单和特性良好，但是计算它们的最好的方法都不明显.

或许，想到的第一个方法是计算特征多项式的系数，再用求根方法求出其根. 遗憾的是，正如第 15 讲所述，这个策略并不好，这是因为即使基础的特征值问题是良态的，由其所导出的多项式求根问题一般也是病态的.（事实上，多项式求根在科学计算中决不是主流课题——主要因为它很少是解决应用问题的最好方法.）

另一个思想是利用如下事实，即在一定条件下，序列

$$\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|},\frac{\boldsymbol{A}\boldsymbol{x}}{\|\boldsymbol{A}\boldsymbol{x}\|},\frac{\boldsymbol{A}^2\boldsymbol{x}}{\|\boldsymbol{A}^2\boldsymbol{x}\|},\frac{\boldsymbol{A}^3\boldsymbol{x}}{\|\boldsymbol{A}^3\boldsymbol{x}\|},\cdots$$

190

收敛于与 $\boldsymbol{A}$ 的绝对值最大的特征值相对应的特征向量. 这一求特征向量的方法称为*幂迭代*（power iteration）. 遗憾的是，虽然幂迭代很著名，但它决不是一般使用的有效方法，除了一些特殊的矩阵外，它的收敛是很慢的.

若不用上述思想，则最通用的特征值算法基于一个不同的原则：计算 $\boldsymbol{A}$ 的显现特征值因子分解，其中特征值作为一个因子的元素出现. 在上一讲中，有 3 个显现特征值的因子分解：对角化，酉对角化和酉三角化（舒尔因子分解）. 在实际计算中，特征值通常是用构建其中一种因子分解来计算的. 概念上，就是要对 $\boldsymbol{A}$ 实施一系列变换，将零引入到必要的位置，正如本书前几讲已讨论过的算法一样. 于是，可以看到求特征值问题与解方程组或最小二乘问题相当类似. 数值线性代数的算法主要是建立一种重复使用的技巧：将零放入到矩阵中.

25.2　基 本 困 难

尽管各种方法互有联系，但计算特征值还是非常特殊的. 其新处在于，代数上的因素决定了任何这类算法都不可能成功.

为了认清这一困难，请注意，正如特征值问题可以化为多项式求根问题一样，反之，任何多项式求根问题也可以表示为特征值问题. 假设有首1多项式

$$p(z)=z^m+a_{m-1}z^{m-1}+\cdots+a_1z+a_0. \tag{25.1}$$

用展成子式的方法，不难证明，$p(z)$等于$(-1)^m$乘以下列$m\times m$矩阵的行列式，

$$\begin{bmatrix} -z & & & & & -a_0 \\ 1 & -z & & & & -a_1 \\ & 1 & -z & & & -a_2 \\ & & 1 & \ddots & & \vdots \\ & & & \ddots & -z & -a_{m-2} \\ & & & & 1 & (-z-a_{m-1}) \end{bmatrix}. \tag{25.2}$$

191

这意味着p的根等于下面矩阵的特征值.

$$A=\begin{bmatrix} 0 & & & & & -a_0 \\ 1 & 0 & & & & -a_1 \\ & 1 & 0 & & & -a_2 \\ & & 1 & \ddots & & \vdots \\ & & & \ddots & 0 & -a_{m-2} \\ & & & & 1 & -a_{m-1} \end{bmatrix}. \tag{25.3}$$

(注意到，如果z是p的根，那么由（25.1）得出，$(1,z,z^2,\cdots,z^{m-1})$是A关于特征值z的左特征向量，利用这一事实，不经（25.2）可以直接得到（25.3).）A称为对应于p的*友矩阵*（companion matrix).

现在计算的困难显现出来了. 众所周知，在多项式系数给定的情况下，任何多项式求根公式是不存在的. 这个不可能性的结果是在19世纪的阿贝尔，伽罗瓦和其他数学家完成的数学工作中的登峰造极成就之一. 在1824年，阿贝尔证明了对于5次和5次以上的多项式不存在类似于二次公式的求根公式.

定理25.1　*对任意$m\geqslant5$，存在次数为m的有理系数的多项式$p(z)$，此多项式有实根r，使得$p(r)=0$，且r不能写成由有理数、加、减、乘、除和k次根组成的表达式.*

这个定理意味着，即使是精确计算，也不可能有计算机程序在有限步内计算出任意多项式的精确根. 由此可见，同样的结论可以应用到计算矩阵特征值的更一般问题.

这并不意味着不能编制一个好的特征值解法，然而，这只是说明，这样的解法不能基于到目前为止用过的求解线性方程组的同类型的技巧. 像豪斯霍尔德反射和高斯消元这样的方法，若可以实行精确的算术运算，则经过有限步后可以精确地解出线性方程组. 相反，

任何特征值解法必须用迭代

特征值解法器的目的是产生快速收敛到特征值的数列，在这点上，特征值计算比解线性方程组更能反映出科学计算的特征，见附录.

最初看来，迭代的要求似乎令人失望，但这一领域内的可用算法收敛相当快. 在大多数情况下，经过一步计算后数列可能有2倍或3倍精确位数的提高. 这样，虽然计算特征值原则上是一个"不可解的"问题，但在实际中，它与解线性代数方程组相比只差一个小的常数因子，此因子一般接近于1而不是10. 理论上说，依赖于$\epsilon_{机器}$的计算量包含了像$\log(|\log(\epsilon_{机器})|)$一样弱的项，见习题25.2. 192

25.3 舒尔因子分解和对角化

如今使用的大多数特征值算法都需要先计算舒尔因子分解. 通过一系列初等酉相似变换 $X \to Q_j^* X Q_j$ 来变换 A，以此来计算舒尔因子分解 $A = QTQ^*$，所以当 $j \to \infty$ 时，乘积

$$\underbrace{Q_j^* \cdots Q_2^* Q_1^*}_{Q^*} A \underbrace{Q_1 Q_2 \cdots Q_j}_{Q} \tag{25.4}$$

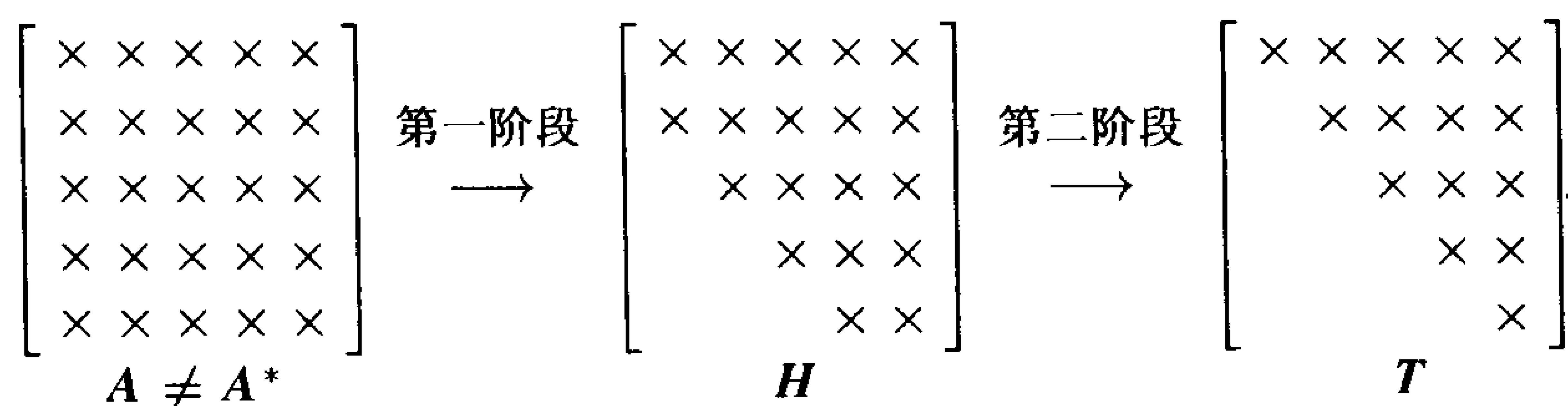
收敛于上三角矩阵 T.

如果 A 是实的非对称矩阵，那么一般来说，矩阵可以有以共轭对出现的复特征值. 这种情况下，舒尔形式是复的，于是，计算舒尔因子分解的算法必须能由实输入产生复输出. 这是可以做到的，毕竟这与实系数多项式求零点有同样性质. 另一方面，如果已经知道要计算的舒尔因子分解为实因子分解，那么整个计算在实算术中是可能的，其中，T 允许沿对角线上有 2×2 块矩阵，每块表示复共轭特征值对. 在实际中，这个选择很重要，并且所有的软件库中都含有这一选择，这里将不详细说明了.

另一方面，假设 A 是埃米尔特矩阵，那么 $Q_j^* \cdots Q_2^* Q_1^* A Q_1 Q_2 \cdots Q_j$ 也是埃米尔特矩阵，于是收敛序列的极限既是三角形矩阵又是埃米尔特矩阵，因此是对角矩阵. 这意味着，计算一般矩阵的酉三角化的算法可以用于计算埃米尔特矩阵酉对角化. 在实际中，虽然在每一步引入了各种修正来利用埃米尔特结构的特殊处理，但本质上还是如何典型地处理埃米尔特问题.

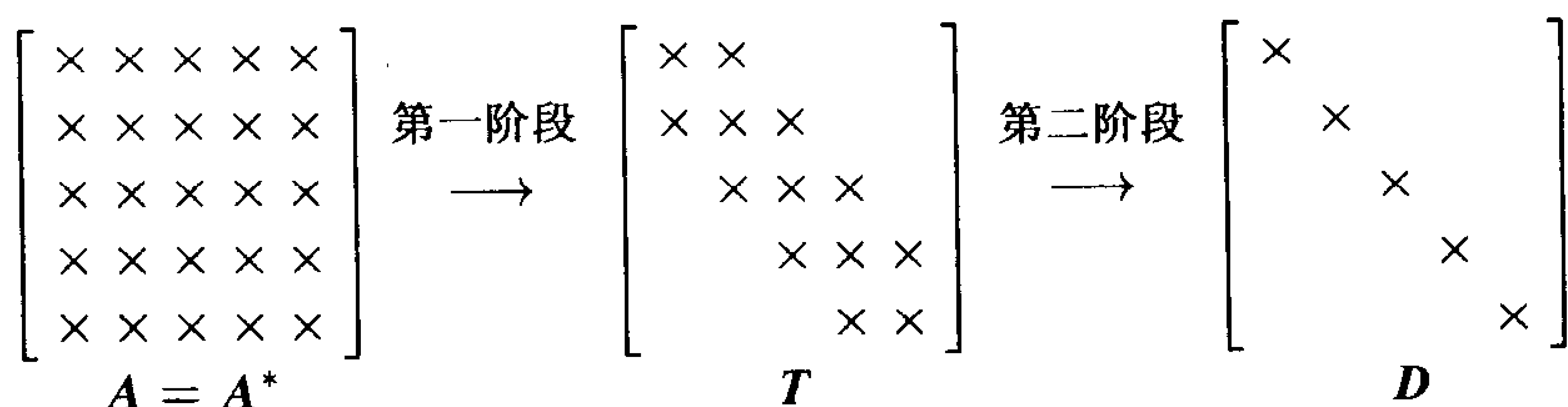
25.4 特征值计算的两个阶段

无论 A 是否是埃米尔特矩阵，计算序列（25.4）经常分为两个阶段. 第1阶段是，运用直接法产生上海森伯格（upper-Hessenberg）矩阵 H，即第一条次对角线以下均为零的矩阵. 第2阶段是，应用迭代法，产生海森伯格矩阵形式上的无限序列，该矩阵序列收敛于三角矩阵. 这个过程图示如下： 193



第 1 阶段，直接约化，需要 $O(m^3)$ 次浮点运算. 第 2 阶段，原则上，迭代阶段不会终止. 因此，如果一直计算下去，将要有无限多次的浮点运算. 然而，在实际计算中，收敛到机器精度要迭代 $O(m)$ 次，每次迭代需要 $O(m^2)$ 次浮点运算，于是，总的计算量为 $O(m^3)$ 次浮点运算. 这些数字说明了第 1 阶段的重要性，没有预备阶段，第 2 阶段的每次迭代将包含满秩矩阵，需要 $O(m^3)$ 的浮点运算，由于达到收敛要迭代 $O(m)$ 次以上，因此总的浮点运算次数为 $O(m^4)$ 或更多.

如果 A 是埃米尔特矩阵，那么两阶段法可能会变得更快. 此时中间矩阵是埃米尔特和海森伯格矩阵，即为三对角（tridiagonal）矩阵，最后的结果是埃米尔特三角矩阵，即如前所述，这是一个对角矩阵. 图示如下：



在埃米尔特矩阵情况下，可以看出，如果仅求特征值（不求特征向量），第 2 阶段的每步仅需 $O(m)$ 浮点运算就可以完成，因而第 2 阶段总的工作量估计为 $O(m^2)$ 浮点运算. 于是，求解埃米尔特矩阵特征值问题，陷入了不合理的情况，算法的“无限”部分不仅在实际上与“有限”部分一样快，而且快了一个数量级.

习　　题

25.1 (a) 设 $A\in\mathbb{C}^{m\times m}$ 是三对角埃米尔特矩阵，其下对角线和上对角线元素均非零. 证明 A 的特征值相异.（提示：证明对任意 $\lambda\in\mathbb{C}$，$A-\lambda I$ 的秩至少为 $m-1$.）

194

(b) 另一方面，设 A 为上海森伯格矩阵，其下对角线元素均非零，给出一个例子，说明 A 的特征值不一定相异.

25.2 设 $e_1,e_2,e_3,\cdots$ 是非负序列，它表示收敛到零的某迭代过程的误差. 假定存在一个常数 C 和一个指数 α，使得对充分大的 k，有 $e_{k+1}\leqslant C\ (e_k)^\alpha$. 特征值计算“第 2 阶段”的不同算法有 3 次收敛(cubic convergence)($\alpha=3$)，平方收敛（quadratic convergence）($\alpha=2$)或线性收敛（linear convergence）（$\alpha=1$ 当 $C<1$），这也称为几何收敛（geometric convergence），这种叫法也许会混淆.

(a) 假设要回答精度 $O(\epsilon_{机器})$，设每步的工作量为 $O(1)$，证明在线性收敛情况下，需要的总工作量为 $O(\log(\epsilon_{机器}))$. 如何把常数 C 放入估计中去？

(b) 证明在超线性收敛，即 $\alpha>1$ 的情况下，所需工作量变为 $O(\log(|\log(\epsilon_{机器})|))$.（提示：这个问题可以用定义新的误差度量 $f_k=C^{1/(\alpha-1)}e_k$ 来化简.）如何把指数 α 放入估计中去？

25.3 设有 3×3 矩阵，希望用酉矩阵 $\boldsymbol{Q}_j$，比如豪斯霍尔德镜射算子或 Givens 旋转来右乘和（或）左乘矩阵，引入零元素. 考虑下面 3 个矩阵结构：

$$(a)\begin{bmatrix}\times&\times&0\\0&\times&\times\\0&0&\times\end{bmatrix},\quad (b)\begin{bmatrix}\times&\times&0\\\times&0&\times\\0&\times&\times\end{bmatrix},\quad (c)\begin{bmatrix}\times&\times&0\\0&0&\times\\0&0&\times\end{bmatrix}.$$

对于每个矩阵，决定下列情况中哪个成立，并证明其结论.

(i) 可以用矩阵 $\boldsymbol{Q}_j$ 左乘的序列来得到.

(ii) 不是(i)，而用矩阵 $\boldsymbol{Q}_j$ 左乘和右乘的序列来得到.

(iii) 用矩阵 $\boldsymbol{Q}_j$ 左乘和右乘的任一序列都不能得到.

第 26 讲　约化到海森伯格型或三对角型

现在讨论在前面一讲中简单提到的两个计算阶段的第 1 阶段：用一系列酉相似变换将满秩矩阵约化为海森伯格阵．如果原来的矩阵是埃米尔特矩阵，那么结果为三对角矩阵．

26.1　一个坏的想法

为计算舒尔因子分解 $A=QTQ^*$，一般想对 A 进行酉相似变换，以便使对角线以下的元素为零．一个自然想法就是用一系列豪斯霍尔德镜射算子把对角线以下的元素相继化为零来直接三角化．

第一个豪斯霍尔德镜射算子 Q_1^* 乘 A 的左边，这将 A 的第 1 列对角线下元素化为零，在这一过程中将改变 A 的所有行．在此及其下面的图中，像通常一样，在每一步改变的元素写成黑体：

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1^*\cdot} \underset{Q_1^*A}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}.$$

196

不幸的是，为完成相似变换，也必须用 Q_1 乘于 A 的右边：

$$\underset{Q_1^*A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\cdot Q_1} \underset{Q_1^*AQ_1}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}.$$

这个过程的效果是，用矩阵所有列的线性组合代替了矩阵的每一列，此过程使得原来引入的零元素被破坏了，最后的结果却没有比开始时更好．

当然，事后认识到这个想法必然会失败，其原因是前一讲讨论过的"基本困难"．有限过程不能精确地求出 A 的特征值．

奇怪的是，如已经讨论过的，这个看起来没有用的非常简单的策略，一般还是有用的，即使不能将对角线以下的元素均化为零，但也可以减少非零元素的个数．在讨论 QR 算法时，将转回到这个"坏的想法"．

26.2　一个好的想法

一个正确的策略是在第一阶段引入零元素时减少一些野心只在更少的矩阵元素上进行操作，即只征服那些一定能保得住的范围.

第一步，先选择不改变第一行的豪斯霍尔德镜射算子 $\boldsymbol{Q}_1^*$，将它左乘 $\boldsymbol{A}$，这形成了第 $2,\cdots,m$ 行的线性组合，将第 1 列的第 $3,\cdots,m$ 行化为零元素. 因此，当 $\boldsymbol{Q}_1$ 右乘 $\boldsymbol{Q}_1^*\boldsymbol{A}$ 时，其第 1 列没有改变，它形成的第 $2,\cdots,m$ 列的线性组合并没有改变已经引入的零元素.

$$\underset{\boldsymbol{A}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\boldsymbol{Q}_1^*\cdot} \underset{\boldsymbol{Q}_1^*\boldsymbol{A}}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{\cdot\boldsymbol{Q}_1} \underset{\boldsymbol{Q}_1^*\boldsymbol{A}\boldsymbol{Q}_1}{\begin{bmatrix} \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}.$$

重复这个想法，在随后的列中引入零元素. 例如，第 2 个豪斯霍尔德镜射算子 $\boldsymbol{Q}_2$，它保持第 1 和第 2 的行和列不变： 197

$$\underset{\boldsymbol{Q}_1^*\boldsymbol{A}\boldsymbol{Q}_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\boldsymbol{Q}_2^*\cdot} \underset{\boldsymbol{Q}_2^*\boldsymbol{Q}_1^*\boldsymbol{A}\boldsymbol{Q}_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{\cdot\boldsymbol{Q}_2} \underset{\boldsymbol{Q}_2^*\boldsymbol{Q}_1^*\boldsymbol{A}\boldsymbol{Q}_1\boldsymbol{Q}_2}{\begin{bmatrix} \times & \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \times & \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}.$$

重复这一过程 $m-2$ 次后，得到了所期望的海森伯格型的一个乘积.

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}$$

$$\underbrace{\boldsymbol{Q}_{m-2}^*\cdots\boldsymbol{Q}_2^*\boldsymbol{Q}_1^*}_{\boldsymbol{Q}^*}\boldsymbol{A}\underbrace{\boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_{m-2}}_{\boldsymbol{Q}}=\boldsymbol{H}.$$

算法写成如下的公式形式. 请把它与算法 10.1 作比较.

算法 26.1　豪斯霍尔德约化到海森伯格形式

for $k=1$ **to** $m-2$

$x = A_{k+1:m,k}$

$v_k = \text{sign}(x_1)\|x\|_2 e_1 + x$

$v_k = v_k / \|v_k\|_2$

$A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2v_k(v_k^* A_{k+1:m,k:m})$

$A_{1:m,k+1:m} = A_{1:m,k+1:m} - 2(A_{1:m,k+1:m}v_k)v_k^*$

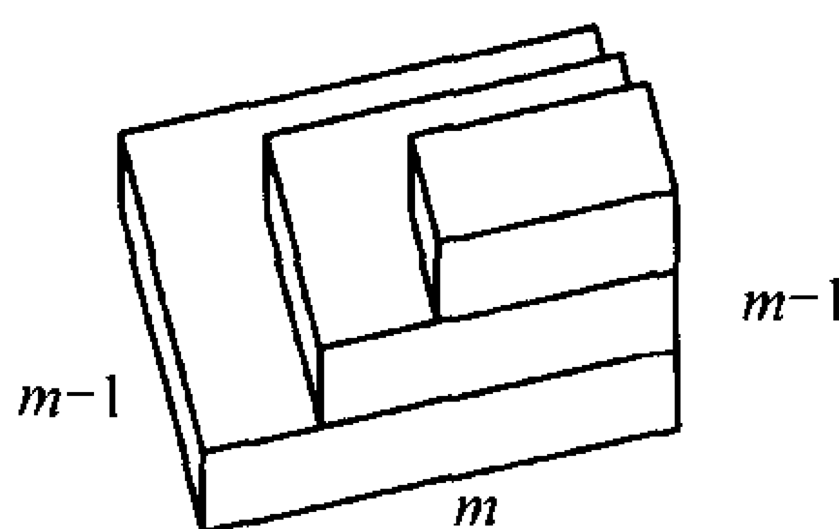
正如算法 10.1，此矩阵 $Q=\prod_{k=1}^{m-2}Q_k$ 不是显式构成的，而反射向量 v_k 却保留下来，可以用 Q 来乘，也可以在需要时重构 Q. 详细内容见第 10 讲.

26.3　运算量估计

算法 26.1 所需的运算量可以用以前用过的几何推理的办法来计算，经验是，酉运算对每个元素需要 4 次浮点运算.

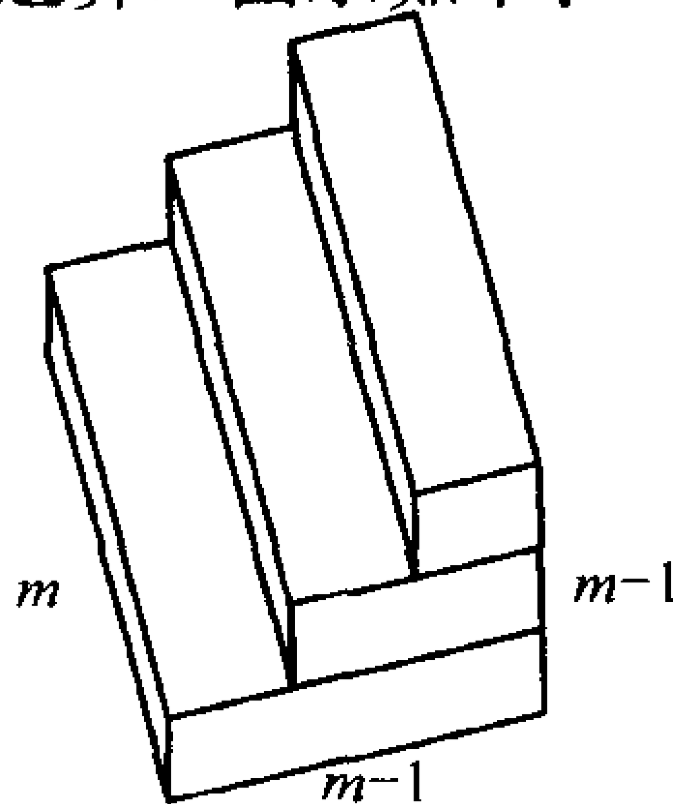
这工作是由对 A 的子矩阵的二次校正来进行的，第一个循环是在矩阵左边应用豪斯霍尔德镜射算子，第 k 个这样的镜射算子应用在最后 $m-k$ 行. 由于前面镜射算子应用的时候，在前 $k-1$ 列中，这些行的元素已为零，所以计算仅在每行的后面 $m-k+1$ 个位置上进行. 图示如下：

198



当 $m\to\infty$ 时，体积收敛于 $\frac{1}{3}m^3$，由于每个元素需要 4 次浮点运算，所以在这个循环中的工作量约为 $\frac{4}{3}m^3$ 次浮点运算.

第二个内循环是在矩阵的右边应用豪斯霍尔德镜射算子，在第 k 步，镜射算子操作通过形成最后 $m-k$ 列的线性组合来实现. 由于零元素不能忽略，所以这个循环包含的工作量比第一个循环要多. 运算必须在每个列的全部 m 个元素上进行，对于单个 k，共有 $m(m-k)$ 个元素参与运算. 图示如下：



当 $m\to\infty$ 时，体积收敛于$\frac{1}{2}m^3$，由于每个元素需要 4 次浮点运算，所以第二个循环约需要 2 m^3次浮点运算.

总共，将 $m\times m$ 矩阵酉约化为海森伯格形式的总工作量为：

$$\text{海森伯格约化工作量：大约为}\frac{10}{3}m^3\text{ 次符点运算.} \qquad (26.1)$$

26.4　埃米尔特情形：约化为三对角型

如果 $\boldsymbol{A}$ 是埃米尔特矩阵，则刚才描述的算法将 $\boldsymbol{A}$ 约化为三对角型（至少，不考虑舍入误差）. 容易看到，由于 $\boldsymbol{A}$ 是埃米尔特矩阵，所以 $\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{Q}$ 也是埃米尔特矩阵，于是任何埃米尔特的海森伯格矩阵是三对角矩阵.

由于零元素在行中引入同样也在列中引入，用忽略这些附加的零元素的方法使一些相关运算可以避免. 利用这一优势，在右边应用豪斯霍尔德镜射算子就变得与在左边应用镜射算子一样简便，应用右镜射算子的总工作量从 $2m^3$ 次浮点运算降为 199 $\frac{4}{3}m^3$ 次. 把两个棱锥相加代替了一个棱锥和一个棱柱，因而运算的总量减至为$\frac{8}{3}m^3$浮点运算.

然而，这计算量的节省仅是基于稀疏性，而非对称性. 事实上，在计算的每个阶段参与运算的矩阵是埃米尔特矩阵. 这给出了两个已经利用了的优点中的一个有利因素，使得总工作量的估计为：

$$\text{三对角约化的工作量：约为}\frac{4}{3}m^3\text{ 次浮点运算.} \qquad (26.2)$$

这里不详细给出运算过程.

26.5　稳　定　性

如同关于 QR 因子分解的豪斯霍尔德算法一样，上面描述的算法是向后稳定的. 回忆定理 16.1，对任意 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$，关于 QR 因子分解的豪斯霍尔德算法计算这些反射向量等价于隐式的、精确酉因子 $\tilde{\boldsymbol{Q}}$（16.2），以及一个显式的上三角形因子 $\tilde{\boldsymbol{R}}$，使得

$$\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{R}}=\boldsymbol{A}+\delta\boldsymbol{A},\qquad \frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}=O(\epsilon_{\text{机器}}).$$

对于算法 26.1，能够建立同类型的误差估计. 设 $\tilde{\boldsymbol{H}}$ 为在浮点运算中计算得到的精确海森伯格矩阵，并设 $\tilde{\boldsymbol{Q}}$，如前一样，对应于用浮点运算来计算的这些反射向量 $\tilde{\boldsymbol{v}}_k$ 的

精确酉矩阵（16.2）. 下面结论能被证明.

定理 26.1　设矩阵$A\in\mathbb{C}^{m\times m}$，其海森伯格约化$A=QHQ^*$可以在满足公理（13.5）和（13.7）的计算机上用算法26.1来计算，并设计算的因子$\tilde{Q}$和$\tilde{H}$按上面指出的来定义，那么对于某一$\delta A\in\mathbb{C}^{m\times m}$，有

$$\tilde{Q}\tilde{H}\tilde{Q}^*=A+\delta A,\quad \frac{\|\delta A\|}{\|A\|}=O(\epsilon_{机器})\tag{26.3}$$

习　　题

26.1　定理26.1和它在后面几讲的后续定理指明，可以在数值上计算A的特征值$\{\tilde{\lambda}_k\}$，是满足条件$\|\delta A\|/\|A\|=O$（$\epsilon_{机器}$）成立的矩阵$A+\delta A$的精确特征值. 是否意味着这些特征值接近于A的精确特征值$\{\lambda_k\}$？这是特征值扰动理论的问题.

200

可以在几何上如下处理上述的问题，给定$A\in\mathbb{C}^{m\times m}$，其谱$\Lambda(A)\subseteq\mathbb{C}$，且给定$\epsilon>0$，定义$A$的2-范数$\epsilon$-伪谱（$\epsilon$-pseudospectrum）$\Lambda_\epsilon(A)$为满足下列条件中任一个条件的数$z\in\mathbb{C}$的集合：

(i) z为$A+\delta A$的一个特征值，其中δA满足$\|\delta A\|_2\leqslant\epsilon$；

(ii) 存在一个向量$u\in\mathbb{C}^m$，满足$\|(A-zI)u\|_2\leqslant\epsilon$和$\|u\|_2=1$；

(iii) $\sigma_m(zI-A)\leqslant\epsilon$；

(iv) $\|(zI-A)^{-1}\|_2\geqslant\epsilon^{-1}$.

在（iv）中的矩阵$(zI-A)^{-1}$称为在z的A的预解式（resolvent）. 如果z为A的特征值，那么用约定$\|(zI-A)^{-1}\|_2=\infty$. 在（iii）中，σ_m表示最小的奇异值.

试证明条件（i）~（iv）是等价的.

26.2　设A为32×32矩阵，其主对角线上的元素为-1，在第一和第二上对角线上的元素为1，其余的元素为零.

(a) 用 MATLAB 或别的软件系统编制 SVD 算法并用曲线绘图软件得到，对于$\epsilon=10^{-1}$，$10^{-2},\cdots,10^{-8}$,A的2-范数ϵ-伪谱边界的曲线.

(b) 对于t，$0\leqslant t\leqslant 50$，画出$\|e^{tA}\|_2$的半对数曲线，在最后衰减开始之前，曲线的初始增长速度是什么？能否把这与伪谱曲线建立联系？（与习题24.3作比较.）

26.3　特征值扰动理论的最著名的结论之一是 Bauer-Fike 定理. 设$A\in\mathbb{C}^{m\times m}$是可对角化的，$A=V\Lambda V^{-1}$，且令$\delta A\in\mathbb{C}^{m\times m}$是任意矩阵，那么$A+\delta A$的每个特征值至少处于中心为$A$的特征值，半径为$\kappa(V)\|\delta A\|_2$的复平面中$m$个圆盘中的一个，其中$\kappa$是2-范数条件数.（与习题24.2作比较.）

(a) 用习题26.1中的等价条件（i）和（iv）证明 Bauer-Fike 定理.

(b) 假设A是正规矩阵，证明，对于$A+\delta A$的每一个特征值$\tilde{\lambda}_j$，存在A的一个特征值λ_j，使得

201

$$|\tilde{\lambda}_j-\lambda_j|\leqslant\|\delta A\|_2.\tag{26.4}$$

第 27 讲　瑞利商，逆迭代

在本讲中，将讨论一些经典的特征值算法. 这些算法适用于不同的情形——尤其是逆迭代，逆迭代是已知特征值求特征向量的标准方法. 不但如此，这些经典方法还是著名的 QR 算法的组成部分，QR 算法将在下面两讲中讨论.

27.1　限于实对称矩阵

在数值线性代数中，大多数算法的思想或应用于一般矩阵，或经某些简化可应用到埃米尔特矩阵. 在本讲及后面 3 讲中，情况也大致如此，但是，一般矩阵与埃米尔特矩阵的一些差别也是相当大的. 因此，在这 4 讲中，为简单起见，将只讨论实对称矩阵. 在这几讲中，还假定$\|\cdot\| = \|\cdot\|_2$.

在这 4 讲中有：$A = A^{\mathrm{T}} \in \mathbb{R}^{m\times m}, x \in \mathbb{R}^m, x^* = x^{\mathrm{T}}, \|x\| = \sqrt{x^{\mathrm{T}}x}$. 实际上，这说明 A 有实的特征值和完全的正交特征向量组. 使用下面记号：

实特征值：$\lambda_1, \cdots, \lambda_m$.

正交特征向量：$q_1, \cdots, q_m$.

202

假定特征向量是由$\|q_j\| = 1$来正规的，若需要，可指定特征值的顺序.

下面几讲中叙述的大部分思想适用于第 25 讲中讨论的两个阶段的第二个阶段，这说明当应用这些思想时，A 不仅是实对称矩阵，而且是三对角矩阵. 三对角结构有时具有数学上的重要性，例如在 QR 算法的位移选择里，以及它总是具有在算法上的重要性，正如本讲末尾所讨论的，把浮点运算的计算量从 $O(m^3)$降至 $O(m)$.

27.2　瑞　利　商

向量 $x \in \mathbb{R}^m$ 的瑞利商（quotient）是标量

$$r(x) = \frac{x^{\mathrm{T}}Ax}{x^{\mathrm{T}}x}. \tag{27.1}$$

注意到，如果 x 是特征向量，那么 $r(x) = \lambda$ 是与之对应的特征值. 欲推导这一公式先思考：给定 x，对此 x 使$\|Ax - \alpha x\|_2$ 最小的什么标量 α "作用上最像一个特征值"？这是形如 $x\alpha \approx Ax$ 的 $m\times 1$ 最小二乘问题（x 是矩阵，α 是未知向量，Ax 是右端项）. 写出这个方程组的法方程（11.9），可得到结果 $\alpha = r(x)$. 于是，当 x 趋于但未必等于特征向量时，$r(x)$就是一个自然的特征值估计.

为使这些想法量化，最好把 $x \in \mathbb{R}^m$ 作为变量，因而 r 是$\mathbb{R}^m \to \mathbb{R}$的函数. 当 x 在

特征向量附近变化时，我们关注 $r(\boldsymbol{x})$ 的局部性质. 研究这个问题的一个方法是求 $r(\boldsymbol{x})$ 关于坐标 x_j 的偏导数：

$$\frac{\partial r(\boldsymbol{x})}{\partial \boldsymbol{x}_j}=\frac{\frac{\partial}{\partial \boldsymbol{x}_j}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x})}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}-\frac{(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x})\frac{\partial}{\partial \boldsymbol{x}_j}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x})}{(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x})^2}$$

$$=\frac{2(\boldsymbol{A}\boldsymbol{x})_j}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}-\frac{(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x})2\boldsymbol{x}_j}{(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x})^2}=\frac{2}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}(\boldsymbol{A}\boldsymbol{x}-r(\boldsymbol{x})\boldsymbol{x})_j.$$

如果把这些偏导数排成一个 m-向量，那么得到 $r(\boldsymbol{x})$ 的梯度（gradient），用 $\nabla r(\boldsymbol{x})$ 来表示. 已经证得

$$\nabla r(\boldsymbol{x})=\frac{2}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}(\boldsymbol{A}\boldsymbol{x}-r(\boldsymbol{x})\boldsymbol{x}). \tag{27.2}$$

从这个式子可以得出，若 $\boldsymbol{x}$ 是 A 的特征向量，那么 $r(\boldsymbol{x})$ 的梯度为零向量. 反之，若 $\nabla r(\boldsymbol{x})=0$ 且 $\boldsymbol{x}\neq 0$，则 $\boldsymbol{x}$ 是特征向量，$r(\boldsymbol{x})$ 是相对应的特征值.

几何上讲，$\boldsymbol{A}$ 的特征向量是函数 $r(\boldsymbol{x})$ 的平稳点（stationary point），$\boldsymbol{A}$ 的特征值是在这些平稳点上 $r(\boldsymbol{x})$ 的值. 事实上，由于 $r(\boldsymbol{x})$ 不依赖于 $\boldsymbol{x}$ 的大小，所以这些平稳点位于过 $\mathbb{R}^m$ 原点的直线上. 如果仅考虑单位球 $\|\boldsymbol{x}\|=1$ 上的平稳点，那么这些平稳点变为孤立点（假设 $\boldsymbol{A}$ 的特征值是单重），见图27-1.

203

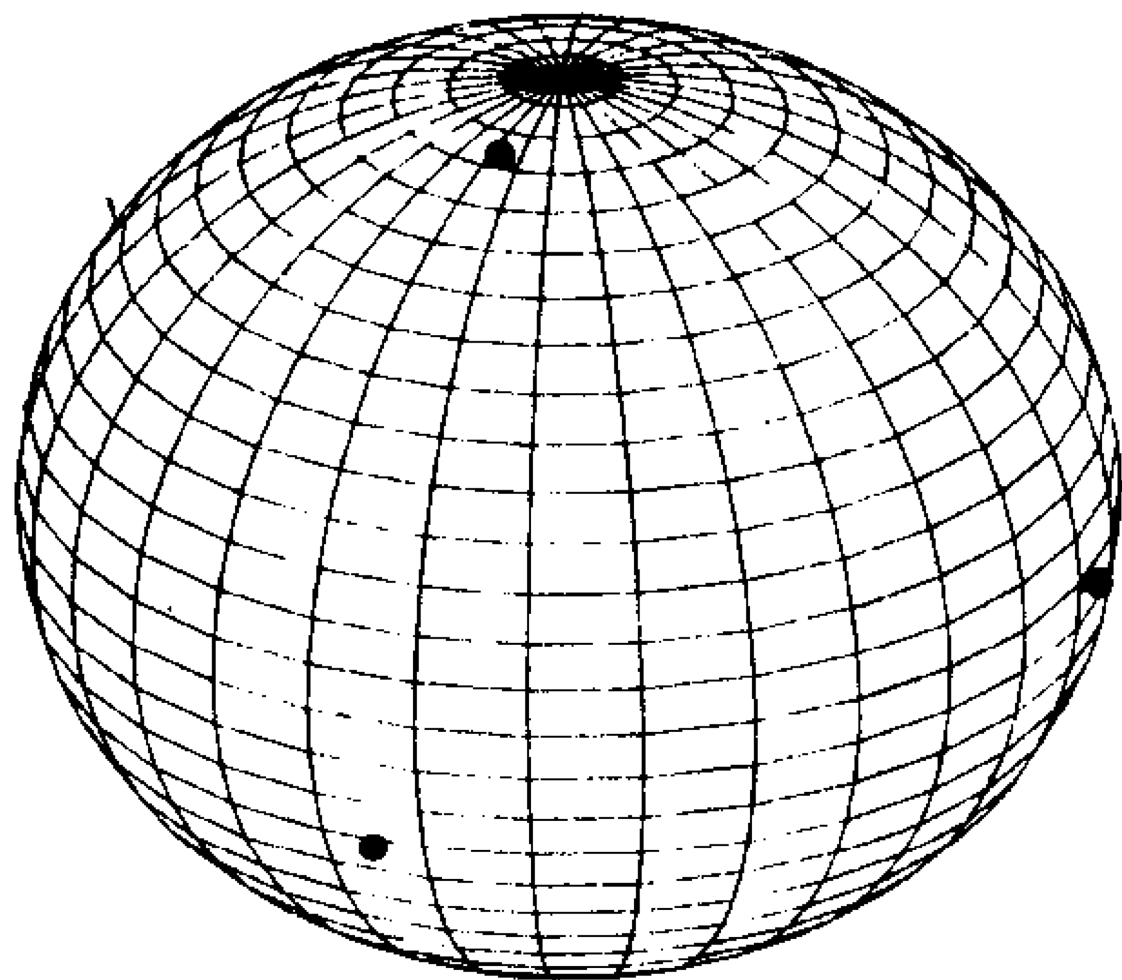


图27-1 瑞利商 $r(\boldsymbol{x})$ 是在 $\mathbb{R}^m$ 中单位球 $\|\boldsymbol{x}\|=1$ 上的连续函数，$r(\boldsymbol{x})$ 的平稳点是 $\boldsymbol{A}$ 的规范化特征向量. 在例子中，$m=3$，有三个正交的平稳点（它们的对径点也正交）

设 $\boldsymbol{q}_j$ 是 $\boldsymbol{A}$ 的一个特征向量. 从 $\nabla r(\boldsymbol{q}_j)=0$ 和函数 $r(\boldsymbol{x})$ 的光滑性（除原点 $\boldsymbol{x}=\boldsymbol{0}$ 外）这一事实出发，可以得出一个重要结论：

$$r(\boldsymbol{x})-r(\boldsymbol{q}_j)=O(\|\boldsymbol{x}-\boldsymbol{q}_j\|^2),\text{当 }\boldsymbol{x}\to\boldsymbol{q}_j. \tag{27.3}$$

于是，瑞利商是特征值的二次精度（quadratically accurate）的估计，此处看出瑞利商的效果.

推导（27.3）的一个更为直接的方法是，把 $\boldsymbol{x}$ 展成 A 的特征向量 $\boldsymbol{q}_1,\cdots,\boldsymbol{q}_m$ 的线性组合. 若 $\boldsymbol{x}=\sum_{j=1}^m a_j\boldsymbol{q}_j$，则 $r(\boldsymbol{x})=\sum_{j=1}^m a_j^2\lambda_j/\sum_{j=1}^m a_j^2$. 于是 $r(\boldsymbol{x})$ 是 A 的特征值的加权平均，其权等于 $\boldsymbol{x}$ 在特征向量基下的坐标的平方. 由于坐标的这个平方，所以不难看出，如果 $|a_j/a_J|\leq\epsilon, j\neq J$，则有 $r(\boldsymbol{x})-r(\boldsymbol{q}_J)=O(\epsilon^2)$.

27.3 幂 迭 代

现在讨论不同的方法，设 $\boldsymbol{v}^{(0)}$ 为一向量，满足 $\|\boldsymbol{v}^{(0)}\|=1$. 下面讨论的幂迭代（power iteration）在第 25 讲开头并未提到它是一个特别好的想法. 可以预料，幂迭代可以得出一个向量序列 $\boldsymbol{v}^{(i)}$，此序列收敛到相应于 A 的最大特征值的特征向量. 204

算法 27.1　幂迭代

$\boldsymbol{v}^{(0)}$ 为 $\|\boldsymbol{v}^{(0)}\|=1$ 的某一向量
for $k=1,2,\cdots$
　$w=A\boldsymbol{v}^{(k-1)}$　　应用 A
　$\boldsymbol{v}^{(k)}=w/\|w\|$　　正规化
　$\lambda^{(k)}=(\boldsymbol{v}^{(k)})^{\mathrm{T}}A\boldsymbol{v}^{(k)}$　　瑞利商

在此和下面的算法中，没有考虑终止条件，而仅用示意的语句“**for** $k=1,2\cdots$”来描述循环. 当然，在实际中，终止条件是非常重要的. 而在这里，恰是像 LAPACK 或 MATLAB 这样一些顶级软件比纯个人编制的程序较为优越的一个地方.

可以容易地分析幂迭代法，把 $\boldsymbol{v}^{(0)}$ 表示为标准正交化特征向量 $\boldsymbol{q}_i$ 的线性组合：

$$\boldsymbol{v}^{(0)}=a_1\boldsymbol{q}_1+a_2\boldsymbol{q}_2+\cdots+a_m\boldsymbol{q}_m.$$

由于 $\boldsymbol{v}^{(k)}$ 是 $A^k\boldsymbol{v}^{(0)}$ 的倍数，所以对某个常数 c_k 有

$$\begin{aligned}\boldsymbol{v}^{(k)}&=c_kA^k\boldsymbol{v}^{(0)}\\&=c_k(a_1\lambda_1^k\boldsymbol{q}_1+a_2\lambda_2^k\boldsymbol{q}_2+\cdots+a_m\lambda_m^k\boldsymbol{q}_m)\\&=c_k\lambda_1^k(a_1\boldsymbol{q}_1+a_2(\lambda_2/\lambda_1)^k\boldsymbol{q}_2+\cdots+a_m(\lambda_m/\lambda_1)^k\boldsymbol{q}_m).\end{aligned}\tag{27.4}$$

由此得如下结论.

定理 27.1　假设 $|\lambda_1|>|\lambda_2|\geq\cdots\geq|\lambda_m|\geq0$ 且 $\boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{v}^{(0)}\neq0$，则当 $k\to\infty$ 时，算法 27.1 的迭代过程满足，

$$\|\boldsymbol{v}^{(k)}-(\pm\boldsymbol{q}_1)\|=O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right),\quad|\lambda^{(k)}-\lambda_1|=O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)\tag{27.5}$$

符号 ± 表示在每步 k，可以选取正号或负号，并且所指明的界成立.

证明　由假定 $a_1 = \boldsymbol{q}_1^{\mathrm{T}}\boldsymbol{v}^{(0)} \neq 0$，所以从（27.4）得第一个等式，第二个等式可以由第一个等式及（27.3）得到. 如果 $\lambda_1 > 0$，那么 ± 号全为 + 或全为 −，反之，如果 $\lambda_1 < 0$，那么 ± 号交替选取. □

在（27.5）和以后相似等式中的 ± 号不好处理，为避免这种复杂性，好的方法 [205] 是用子空间收敛而不是向量收敛，例如，不是 $\langle \boldsymbol{v}^{(k)} \rangle$ 收敛到 $\langle \boldsymbol{q}_1 \rangle$，然而，为了避免陷入讨论子空间的收敛细节，将不讨论这种收敛性.

由于一些原因，幂迭代本身在应用上是有限的. 首先，幂迭代只能用于计算与最大特征值对应的特征向量. 其次，收敛是线性的，每次迭代仅以常数因子约等于 $|\lambda_2/\lambda_1|$ 来减少误差. 最后，这个因子的优劣依赖于是否有比其他特征值大很多的最大特征值. 如果最大的两个特征值在量值上很接近，那么收敛非常慢.

幸运的是，有办法可以扩大特征值之间的差别.

27.4　逆　迭　代

对任意 $\mu \in \mathbb{R}$，μ 不是 A 的特征值，$(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 的特征向量与 $\boldsymbol{A}$ 的特征向量是相同的，相应的特征值是 $\{(\lambda_j - \mu^{-1})\}$，其中 $\{\lambda_j\}$ 是 $\boldsymbol{A}$ 的特征值. 这个结论提示了一个想法，假设 μ 是靠近 $\boldsymbol{A}$ 的一个特征值 λ_J，那么 $(\lambda_J - \mu)^{-1}$ 可以比 $(\lambda_j - \mu)^{-1}$，$j \neq J$ 的全部指标大很多. 于是，如果对 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 应用幂迭代，那么这个过程将很快收敛于 $\boldsymbol{q}_J$. 这一想法称为*逆迭代*（inverse iteration）.

算法 27.2　**逆迭代**	
$\boldsymbol{v}^{(0)}$ 等于 $\\|\boldsymbol{v}^{(0)}\\| = 1$ 的某一向量	
for $k = 1, 2, \cdots$	
Solve $(A-\mu\boldsymbol{I})w = \boldsymbol{v}^{(k-1)}$ for w	应用 $(A-\mu\boldsymbol{I})^{-1}$
$\boldsymbol{v}^{(k)} = w/\\|w\\|$	正规化
$\lambda^{(k)} = (\boldsymbol{v}^{(k)})^{\mathrm{T}} A \boldsymbol{v}^{(k)}$	瑞利商

如果 μ 是 $\boldsymbol{A}$ 的特征值，以致 $\boldsymbol{A}-\mu\boldsymbol{I}$ 是奇异的，怎么办？如果 μ 近似于 $\boldsymbol{A}$ 的一个特征值，以致 $\boldsymbol{A}-\mu\boldsymbol{I}$ 是病态的，从而导致不能得到 $(\boldsymbol{A}-\mu\boldsymbol{I})\ \boldsymbol{w} = \boldsymbol{v}^{(k-1)}$ 的准确解怎么办？逆迭代的这些表面上的陷阱完全不是问题. 见习题 27.5.

与幂迭代一样，逆迭代也仅是线性收敛，然而与幂迭代不同，可以用给出特征值的估计 μ 来求出相应的特征向量，另外，由于线性收敛的速度依赖于 μ 的优劣，所以是可以控制的. 如果 μ 与 $\boldsymbol{A}$ 的一个特征值比 μ 与 $\boldsymbol{A}$ 的其余特征值近得多，那么 $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$ 的最大特征值比其他特征值大得多. 与幂迭代相同的原因，可以得出下面的定理.

[206] **定理 27.2**　*设* λ_J *是最靠近* μ *的特征值.* λ_K *是第二个最靠近* μ *的特征值，即*

$|\mu-\lambda_J|<|\mu-\lambda_K|\leq|\mu-\lambda_j|, \forall j\neq J$，并且假定 $q_J^{\mathrm{T}}v^{(0)}\neq 0$，则算法 27.2 的迭代，当 $k\to\infty$ 时，满足

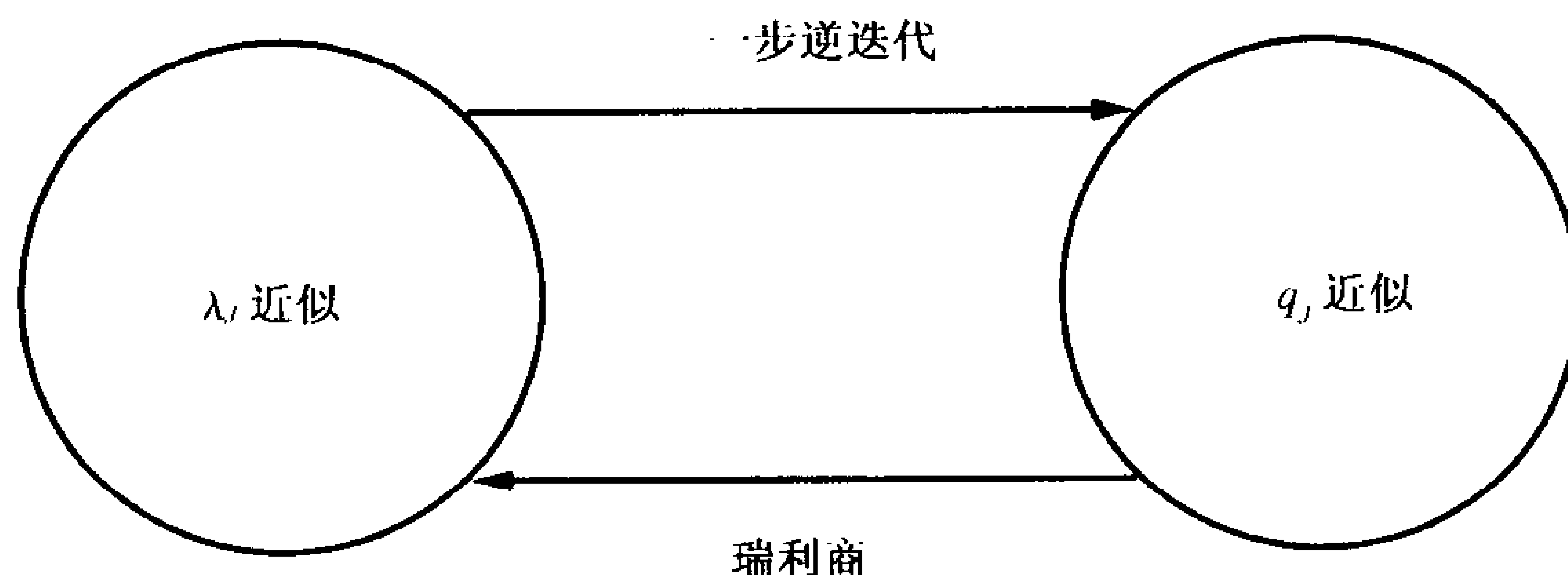
$$\|v^{(k)}-(\pm q_J)\|=O\left(\left|\frac{\mu-\lambda_J}{\mu-\lambda_K}\right|^k\right),\quad |\lambda^{(k)}-\lambda_J|=O\left(\left|\frac{\mu-\lambda_J}{\mu-\lambda_K}\right|^{2k}\right)$$

其中符号 ± 与定理 27.1 中意义的一样.

如果矩阵的特征值已知，逆迭代是计算一个或多个特征向量的标准方法，因此它是数值线性代数中最有价值的工具之一. 在此情况下，算法 27.2 按书写的那样使用，只是其中瑞利商的计算可以删除掉.

27.5 瑞利商迭代

到这讲为止，已经讨论了从特征向量的估计求得特征值估计的方法（瑞利商）和从特征值估计得到特征向量估计的另一方法（逆迭代）. 可以把这两个方法结合起来：



（这图过于简单了：为从 λ_J 的近似经一步逆迭代得到 q_J 的近似，先需要对 q_J 有一个初步的近似.）这个思路就是用不断改进特征值估计来提高每一步逆迭代的收敛速度. 这个算法称为瑞利商迭代（Rayleigh quotient iteration）.

算法 27.3 瑞利商迭代

$v^{(0)}=\|v^{(0)}\|=1$ 的某一向量

$\lambda^{(0)}=(v^{(0)})^{\mathrm{T}}Av^{(0)}=$ 相应的瑞利商

for $k=1,2,\cdots$

  Solve $(A-\lambda^{(k-1)}I)w=v^{(k-1)}$ for w 应用 $(A-\lambda^{(k-1)}I)^{-1}$

  $v^{(k)}=w/\|w\|$ 正规化

  $\lambda^{(k)}=(v^{(k)})^{\mathrm{T}}Av^{(k)}$ 瑞利商

207

该算法的收敛令人吃惊：每一步迭代后精确数字的位数可增至 3 倍.

定理 27.3 初始向量 $v^{(0)}$ 任取于非零测集，则瑞利商迭代收敛于一对特征值或特征向量. 当方法收敛时，这收敛最终是 3 次收敛，即如果 λ_J 是 A 的特征值，$v^{(0)}$ 充分接近特征向量 q_J，那么当 $k\to\infty$ 时，有

$$\|\boldsymbol{v}^{(k+1)}-(\pm \boldsymbol{q}_J)\| = O(\|\boldsymbol{v}^{(k)}-(\pm \boldsymbol{q}_J)\|^3) \tag{27.6}$$

$$|\lambda^{(k+1)}-\lambda_J| = O(|\lambda^{(k)}-\lambda_J|^3) \tag{27.7}$$

其中，(27.6) 两边的±号未必相同.

证明　我们不证明对几乎所有初始向量都收敛的论断，在此只证明，如果方法收敛，那么收敛最终是3阶的. 为简化起见，假定特征值 λ_J 是简单特征值. 利用 (27.3)，如果对于充分小的ϵ，有 $\|\boldsymbol{v}^{(k)}-q_J\|\leq\epsilon$，那么瑞利商得到特征值估计 $\lambda^{(k)}$，并有 $|\lambda^{(k)}-\lambda_J|=O(\epsilon^2)$. 应用证明定理27.2时的论证，如果由 $\boldsymbol{v}^{(k)}$ 和 $\lambda^{(k)}$ 实施一步逆迭代得到新的 $\boldsymbol{v}^{(k+1)}$，那么

$$\|\boldsymbol{v}^{(k+1)}-q_J\| = O(|\lambda^{(k)}-\lambda_J|\|\boldsymbol{v}^{(k)}-q_J\|) = O(\epsilon^3)\ .$$

此外，在 λ_J 和 $\boldsymbol{q}_J$ 的充分小的领域内，隐含在符号 O 中的常数是一致的. 于是，得到了下面模式的收敛性：

$$\begin{array}{ccc}
\|\boldsymbol{v}^{(k)}-(\pm\boldsymbol{q}_J)\| & & |\lambda^{(k)}-\lambda_J| \\
\epsilon & \to & O(\epsilon^2) \\
\downarrow & \swarrow & \\
O(\epsilon^3) & \to & O(\epsilon^6) \\
\downarrow & \swarrow & \\
O(\epsilon^9) & \to & O(\epsilon^{18}) \\
\vdots & & \vdots
\end{array}$$

由刚才提到的一致性可得估计 (27.6) ~ (27.7). □

例27.1　3阶收敛是如此快，以至须用数值例子来说明. 考虑对称矩阵

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{bmatrix},$$

208

设 $\boldsymbol{v}^{(0)} = (1,\ 1,\ 1)^{\mathrm{T}}/\sqrt{3}$为初始向量估计，当瑞利商迭代应用到$\boldsymbol{A}$时，用最初的3次迭代计算下面的 $\lambda^{(k)}$：

$$\lambda^{(0)} = 5,\quad \lambda^{(1)} = 5.2131\cdots,\quad \lambda^{(2)} = 5.214319743184\cdots.$$

最接近 $\boldsymbol{v}^{(0)}$ 的特征向量所对应的特征值的真实值为 $\lambda = 5.214319743377$. 仅3次迭代后，瑞利商迭代得到的结果精确到10位数字. 若计算机精确度足够高，那么再迭代3次后，其精度将到270位. □

27.6　运算量估计

在本讲的最后，将讨论已经叙述过的3个迭代中实施每步所需的工作量.

首先，假设 $A \in \mathbb{R}^{m\times m}$ 是满矩阵，幂迭代每步包含了矩阵与向量的乘法，需要 $O(m^2)$ 浮点运算，逆迭代的每步包含了解方程组，需要 $O(m^3)$ 浮点运算，但如果事先把矩阵用 LU 分解或 QR 分解或其他方法进行处理，那么运算数就降为 $O(m^2)$. 在瑞利商迭代情况中，每步矩阵需要求逆变换，因此每步的浮点运算控制在 $O(m^3)$ 不是很容易的.

如果 A 是三对角矩阵，那么所需计算量就会大大减少，3 个迭代的每步只需 $O(m)$ 浮点运算. 附带地，对于涉及非对称矩阵的类似迭代，必须处理海森伯格矩阵而不是三对角矩阵，因此其计算量增加到 $O(m^2)$.

习　题

27.1　设 $A \in \mathbb{C}^{m\times m}$ 给定，不必是埃米尔特矩阵，试证明，数 $z \in \mathbb{C}$ 是 A 的瑞利商的充分必要条件是，它是关于某一酉矩阵 Q 的 Q^*AQ 的对角线元素. 于是，若正交变换到右手坐标系，瑞利商刚好是矩阵的对角线元素.

27.2　令 $A \in \mathbb{C}^{m\times m}$ 为任一矩阵，对任意非零向量 $x \in \mathbb{C}^m$，A 有相应的瑞利商. A 的所有瑞利商集合称为值域或 A 的数值值域，用 $W(A)$ 表示复平面上的这个子集.

(a) 证明 $W(A)$ 是包含 A 的特征值的凸包，(b) 证明，若 A 为正规矩阵，那么 $W(A)$ 等于 A 的特征值的凸包.

27.3　证明，对于非埃米尔特矩阵 $A \in \mathbb{C}^{m\times m}$，瑞利商 $r(x)$ 给出了特征值估计，其精确度一般是线性的，不是二次的. 试说明，这提示了关于非埃米尔特矩阵的瑞利商的收敛速度. 209

27.4　每个实对称方阵能正交对角化以及本讲的结论在正交坐标变换下是不变的，于是，实现本讲的每个推导仅假定 A 为对角矩阵并在对角线上元素按绝对值递减排列就足够了. 作了这些假定后，试叙述 (27.4)，(27.5) 的形式和算法 27.3.

27.5　如在课本中提及的，逆迭代依赖于方程组求解，而此方程组是可以有条件数阶为 $\epsilon_{机器}^{-1}$ 的非常病态的方程组. 众所周知，一般不可能精确地求解病态方程组. 这不是在算法中的致命缺点吗？

回答是否定的，即在逆迭代中病态不是问题，此证明如下. 假设 A 是一个实对称矩阵，有一个特征值，其绝对值远小于其他特征值（不失一般性，可设 $\mu=0$）. 设 v 是一个向量，其分量在 A 的所有特征向量 $q_1,\cdots,q_m$ 方向上，并假定 $Aw=v$ 可以稳定地向后求解，得到计算的向量 $\tilde{w}$. 使用的计算指出，即使 $\tilde{w}$ 可以与 w 相差很大，$\tilde{w}/\|\tilde{w}\|$ 与 $w/\|w\|$ 仍很接近. 210

27.6　如果 A 的两个特征值相等，那么图 27-1 会发生什么？

第 28 讲　无位移的 QR 算法

起始于 20 世纪 60 年代初期的 QR 算法是数值分析的一颗明珠. 这里将介绍最简单形式的 QR 算法，这个算法可以看作计算矩阵幂 $\boldsymbol{A},\boldsymbol{A}^2,\boldsymbol{A}^3,\cdots$ 的 QR 分解的一个稳定方法.

28.1　QR 算法

QR 算法的最基本的形式看起来也不简单.

算法 28.1　纯 QR 算法	
$A^{(0)}=A$	
for $k=1,2,\cdots$	
$Q^{(k)}R^{k}=A^{(k-1)}$	$A^{(k-1)}$ 的 QR 因子分解
$A^{(k)}=R^{(k)}Q^{(k)}$	以逆序重新组合因子

所需要做的就是进行 QR 因子分解，再用相反次序将得到的因子 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 相乘，再重复这一过程. 在适当的假设下，如果 $\boldsymbol{A}$ 是任意矩阵，那么该简单算法收敛到 $\boldsymbol{A}$ 的舒尔型——上三角形矩阵，如果 $\boldsymbol{A}$ 为埃米尔特矩阵，那么算法收敛到对角矩阵. 为讨论简单起见，将保留上一讲的假设，即 $\boldsymbol{A}$ 为具有实特征值 λ_j 和标准正交特征向量 $\boldsymbol{q}_j$ 的实对称矩阵. 于是，仅讨论矩阵 $\boldsymbol{A}^{(k)}$ 收敛到对角型矩阵.

211

收敛到对角型矩阵可用于求出 $\boldsymbol{A}$ 的特征值，当然，中间所涉及的变换必须是相似变换. 容易证明：QR 算法先对 $\boldsymbol{A}^{(k)}$ 三角形化，方法如下：令 $\boldsymbol{R}^{(k)}=(\boldsymbol{Q}^{(k)})^{\mathrm{T}}\boldsymbol{A}^{(k-1)}$，在右边乘 $\boldsymbol{Q}^{(k)}$，得到 $\boldsymbol{A}^{(k)}=(\boldsymbol{Q}^{(k)})^{\mathrm{T}}\boldsymbol{A}^{(k-1)}\boldsymbol{Q}^{(k)}$. 事实上，以前已出现过这个相似变换：在第 26 讲中提到的是一个“坏想法”，若想仅用一步就试图把 $\boldsymbol{A}$ 化成三角形形式，那么这个变换是一个坏的想法，但把这个变换作为迭代的基础，那么它将是十分有用的.

如瑞利商迭代一样，对于实对称矩阵，QR 算法也是 3 阶收敛的. 然而，为达到这个效果，上面的算法必须在每步引入位移来进行修改. 使用位移是算法 28.1 的 3 个修改之一，修改后将更贴近于实际应用的算法.

1. 开始迭代之前，先将 $\boldsymbol{A}$ 化为三对角矩阵，做法如第 26 讲所讨论的.
2. 每步不是对 $\boldsymbol{A}^{(k)}$ 而是对位移矩阵 $\boldsymbol{A}^{(k)}-\mu^{(k)}\boldsymbol{I}$ 做因子分解，其中 $\boldsymbol{\mu}^{(k)}$ 是某个特征值估计.
3. 只要有可能，特别是当找到一个特征值，那么就用 $\boldsymbol{A}^{(k)}$ 分裂为子矩阵形式，

从而把问题“压缩”了.

下面简要写出结合了这些修改的 QR 算法.

算法 28.2　“实际的”QR 算法

$(Q^{(0)})^{\mathrm{T}}A^{(0)}Q^{(0)}=A$　　　　$A^{(0)}$ 是 A 的一个对角化

for $k=1,2,\cdots$

　　选择位移 $\mu^{(k)}$　　　　例如，选取 $\mu^{(k)}=A_{mm}^{(k-1)}$

　　$Q^{(k)}R^{(k)}=A^{(k-1)}-\mu^{(k)}I$　　　　$A^{(k-1)}-\mu^{(k)}I$ 的 QR 因子分解

　　$A^{(k)}=R^{(k)}Q^{(k)}+\mu^{(k)}I$　　　　此逆序重新组合因子

　　如果任一非对角元素 $A_{j,j+1}^{(k)}$ 充分趋近于 0,

　　　　令 $A_{j,j+1}=A_{j+1,j}=0$，得到

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}=A^{(k)}$$

　　　　现在对 A_1 和 A_2 应用 QR 算法.

具有恰当位移的 QR 算法，自 20 世纪 60 年代初期以来就是计算矩阵全部特征值的标准方法. 仅在 20 世纪 90 年代出现了竞争者，即在第 30 讲中将讨论的分而治之算法.

在第 26 讲中已讨论了三对角化，在下一讲中将讨论位移，而在本书中将不深入讨论压缩. 现在只将精力放在“纯”QR 算法和说明怎样求特征值上. 212

28.2　非正规化同时迭代

这里讨论的是与 QR 算法相关的另一种算法，称其为*同时迭代*（simultaneous iteration），其性质更为明显.

同时迭代想法是把幂代迭法同时应用到几个向量上.（其等价术语为*块幂迭代*（block power iteration）.）假设以由 n 个线性无关的向量 $\boldsymbol{v}_1^{(0)},\cdots,\boldsymbol{v}_n^{(0)}$ 组成的集合为迭代的开始. 有理由相信，当 $k\to\infty$ 时，$A^k\boldsymbol{v}_1^{(0)}$（在适当的假设下）收敛到对应于 A 的绝对值最大的特征值的特征向量. 空间 $\langle A^k\boldsymbol{v}_1^{(0)},\cdots,A^k\boldsymbol{v}_n^{(0)}\rangle$ 应收敛（在适当的假设下）到由 A 的特征向量 $\boldsymbol{q}_1,\cdots,\boldsymbol{q}_n$ 所张成的空间 $\langle\boldsymbol{q}_1,\cdots,\boldsymbol{q}_n\rangle$，其中这 n 个特征向量对应于 n 个依绝对值最大的特征值.

用矩阵记号，可如下进行，定义 $V^{(0)}$ 为 $m\times n$ 初始矩阵，

$$V^{(0)}=\left[\begin{array}{c|c|c} \boldsymbol{v}_1^{(0)} & \cdots & \boldsymbol{v}_n^{(0)} \end{array}\right], \tag{28.1}$$

并定义 $V^{(k)}$ 为 A 的 k 次应用后的结果

$$V^{(k)}=A^kV^{(0)}=\left[\begin{array}{c|c|c} v_1^{(k)} & \cdots & v_n^{(k)} \end{array}\right].\tag{28.2}$$

由于关注的是 $V^{(k)}$ 的列向量空间，所以可以通过计算 $V^{(k)}$ 的约化 QR 因子分解：

$$\hat{Q}^{(k)}\hat{R}^{(k)}=V^{(k)}.\tag{28.3}$$

来构造这个空间的一组优质基，其中 $\hat{Q}^{(k)}$ 和 $\hat{R}^{(k)}$ 分别是阶为 $m\times n$ 和 $n\times n$ 的矩阵．在一定条件下，当 $k\to\infty$ 时，$\hat{Q}^{(k)}$ 的顺序列向量似乎可能收敛到特征向量 $\pm q_1,\pm q_2,\cdots,\pm q_n$.

这个推测可以用类似于上一讲的分析来证明．如果把 $v_j^{(0)}$ 和 $v_j^{(k)}$ 按照 A 的特征向量展开，有

$$v_j^{(0)}=a_{1j}q_1+\cdots+a_{mj}q_m,$$
$$v_j^{(k)}=\lambda_1^ka_{1j}q_1+\cdots+\lambda_m^ka_{mj}q_m.$$

与上一节一样，若两个条件都满足时，则收敛结论成立．第一个假设为，前 $n+1$ 个特征值按照绝对值是互不同的：

213

$$|\lambda_1|>|\lambda_2|>\cdots>|\lambda_n|>|\lambda_{n+1}|\geqslant|\lambda_{n+2}|\geqslant\cdots\geqslant|\lambda_m|.\tag{28.4}$$

第二个假设是，展式系数 a_{ij} 的矩阵在适当的意义下是非奇异的．令 $\hat{Q}$ 为 $m\times n$ 矩阵，其列是特征向量 $q_1,q_2,\cdots,q_n$.（特征向量的矩阵 $\hat{Q}$ 完全不同于约化 QR 因子分解中的因子 $\hat{Q}^{(k)}$）我们作如下假定：

$$\hat{Q}^{\mathrm{T}}V^{(0)}\text{的所有前主子阵是非奇异的.}\tag{28.5}$$

$Q^{\mathrm{T}}V^{(0)}$ 的前主子阵，其意义是阶为 $1\times1,2\times2,\cdots,n\times n$ 的左上方子阵．（条件(28.5) 等价于 $\hat{Q}V^{(0)}$ 有 LU 因子分解的条件；见习题 20.1）

定理 28.1　*假设迭代 (28.1)～(28.3) 可以实现，并且条件 (28.4) 和 (28.5) 均满足，则当 $k\to\infty$ 时，矩阵 $\hat{Q}^{(k)}$ 线性收敛到 A 的特征向量：对于每个 j，$1\leqslant j\leqslant n$，有*

$$\|q_j^{(k)}-\pm q_j\|=O(C^k)\tag{28.6}$$

其中，$C<1$ 为常数 $\max_{1\leqslant k\leqslant n}|\lambda_{k+1}|/|\lambda_k|$，如在上一讲的定理中一样，符号 ± 表示在第 k 步上，正号或负号均可取，并且所指明的界成立.

证明　将 $\hat{Q}$ 扩展为 A 的特征向量的 $m\times m$ 的正交满矩阵，令 Λ 为相应特征值的对角矩阵，于是有 $A=Q\Lambda Q^{\mathrm{T}}$．如刚才提到的，$\hat{Q}$ 是 Q 的前 $m\times n$ 部分一样，定义 $\hat{\Lambda}$（仍为对

角矩阵）是 Λ 的前 $n\times n$ 部分，那么当 $k\to\infty$ 时，有

$$V^{(k)}=A^kV^{(0)}=Q\Lambda^kQ^{\mathrm{T}}V^{(0)}=\hat{Q}\hat{\Lambda}^k\hat{Q}^{\mathrm{T}}V^{(0)}+O(|\lambda_{n+1}|^k)$$

如果（28.5）成立，特别地，有 $\hat{Q}^{\mathrm{T}}V^{(0)}$ 非奇异，所以用 $(\hat{Q}^{\mathrm{T}}V^{(0)})^{-1}\hat{Q}^{\mathrm{T}}V^{(0)}$ 右乘 $O(|\lambda_{n+1}|^k)$，将方程变为

$$V^{(k)}=(\hat{Q}\hat{\Lambda}^k+O(|\lambda_{n+1}|^k))\hat{Q}^{\mathrm{T}}V^{(0)}.$$

由于 $\hat{Q}^{\mathrm{T}}V^{(0)}$ 是非奇异的，所以这个矩阵的列空间与

$$\hat{Q}\hat{\Lambda}^k+O(|\lambda_{n+1}|^k).$$

的列空间相同. 从 $\hat{Q}\hat{\Lambda}^k$ 的形式和假设（28.4），显然，这个列空间线性收敛到 $\hat{Q}$ 的列空间. 这个收敛性可以定量化，例如可以用定义子空间之间的夹角来实现，详细证明在此省略.

事实上，不仅仅假定 $\hat{Q}^{\mathrm{T}}V^{(0)}$ 是非奇异的，而且还假定它的所有前主子阵是非奇异的. 由此可见，上面的证明也可以应用到 $V^{(k)}$ 和 $\hat{Q}$ 的列向量的前子集：第一个列向量，第一个和第二个列向量，第一个，第二个和第三个列向量，等等. 在每一种情形，由 $V^{(k)}$ 的已指出的列向量张成的空间，线性收敛到 $\hat{Q}$ 的相应列向量张成的空间，从全部逐次列空间收敛性和 QR 因子分解（28.3）的定义，可以得到（28.6）. □ 214

28.3 同时迭代

当 $k\to\infty$ 时，在算法（28.1）~（28.3）中的向量 $\boldsymbol{v}_1^{(k)},\cdots,\boldsymbol{v}_n^{(k)}$ 均收敛到 $\boldsymbol{A}$ 的同一个主特征向量 $\boldsymbol{q}_1$ 的倍数. 于是，尽管它们张成的空间 $\langle\boldsymbol{v}_1^{(k)},\cdots,\boldsymbol{v}_j^{(k)}\rangle$ 可收敛到某些有用的向量，但这些向量构成了高度病态的空间基. 如果用浮点运算来实现刚才讨论的同时迭代，由于舍入误差的缘故，所需信息会很快地失去.

改进的办法是简单的：必须在每步进行标准正交化而不是仅做一次标准正交化. 于是，将不构造如上定义的 $V^{(k)}$，而构造具有相同列空间的不同的矩阵序列 $Z^{(k)}$.

算法 28.3　同时迭代

取 $\hat{Q}^{(0)}\in\mathbb{R}^{m\times n}$ 为正交列.
for $k=1,2,\cdots$
　　$Z=A\hat{Q}^{(k-1)}$
　　$\hat{Q}^{(k)}\hat{R}^{(k)}=Z$　　　Z 的约化 QR 因子分解

从算法的构成上，显然，$\hat{Q}^{(k)}$ 和 $Z^{(k)}$ 的列空间是一样的，它们都等于 $A^k\hat{Q}^{(0)}$ 的列空

间. 从数学上看，同时迭代的新公式收敛条件与原来的相同.

定理 28.2　算法 28.3 得到的矩阵$\hat{Q}^{(k)}$与定理 28.1 中迭代（28.1）~（28.3）产生的矩阵一样（假定初始矩阵是$\hat{Q}^{(0)}$是相同的），在相同假设（28.4）和（28.5）下，其收敛性与定理 28.1 中所叙述的一样.

28.4　同时迭代⟺QR 算法

可以对 QR 算法作如下说明，QR 算法等价于应用到 $n=m$ 初始向量的全集，即单位阵，$\hat{Q}^{(0)}=I$ 的同时迭代. 由于矩阵$\hat{Q}^{(k)}$为方矩阵，所以应处理完全 QR 因子分解并可去掉$\hat{Q}^{(k)}$和$\hat{R}^{(k)}$上的帽子. 事实上，将用$R^{(k)}$来代替$\hat{R}^{(k)}$，但为了区分同时迭代的 Q 矩阵与 QR 分解中的 Q 矩阵，将用$\underline{Q}^{(k)}$来代替$\hat{Q}^{(k)}$.

215

下面为确定以$\underline{Q}^{(0)}=I$ 的同时迭代的 3 个公式，接下来是取作 $m\times m$ 矩阵 $A^{(k)}$ 的定义的第 4 个公式：

同时迭代：

$$\underline{Q}^{(0)} = I \tag{28.7}$$
$$Z = A\underline{Q}^{(k-1)}, \tag{28.8}$$
$$Z = \underline{Q}^{(k)}R^{(k)}, \tag{28.9}$$
$$A^{(k)} = (\underline{Q}^{(k)})^{\mathrm{T}}A\underline{Q}^{(k)}. \tag{28.10}$$

在下面为确定纯 QR 算法的 3 个公式，紧接着取作 $m\times m$ 矩阵$\underline{Q}^{(k)}$的定义的第 4 个公式.

无位移的 QR 算法

$$A^{(0)} = A, \tag{28.11}$$
$$A^{(k-1)} = Q^{(k)}R^{(k)}, \tag{28.12}$$
$$A^{(k)} = R^{(k)}Q^{(k)}, \tag{28.13}$$
$$\underline{Q}^{(k)} = Q^{(1)}Q^{(2)}\cdots Q^{(k)}. \tag{28.14}$$

另外，对于这两种算法，进而定义 $m\times m$ 矩阵$\underline{R}^{(k)}$，

$$\underline{R}^{(k)}=R^{(k)}R^{(k-1)}\cdots R^{(1)}. \tag{28.15}$$

下面将给出两个算法的等价性.

定理 28.3　算法（28.7）~（28.10）和（28.11）~（28.14）得到了恒等的矩阵序列$\underline{R}^{(k)}$，$\underline{Q}^{(k)}$和 $A^{(k)}$，即它们由 A 的第 k 次幂的 QR 因子分解，

$$A^k=\underline{Q}^{(k)}\underline{R}^{(k)} \tag{28.16}$$

以及投影

$$A^{(k)} = (\underline{Q}^{(k)})^{\mathrm{T}} A \underline{Q}^{(k)}. \tag{28.17}$$

来确定.

证明 对 k 作归纳证明. 对于 $k=0$ 情况是明显的，对于同时迭代和 QR 算法，方程（28.7）~（28.15）可推出 $A^0 = \underline{Q}^{(0)} = \underline{R}^{(0)} = I$ 和 $\underline{A}^{(0)} = A$. 由此直接可得到（28.6）和（28.17）.

216

现考虑关于同时迭代 $k \geqslant 1$ 情形. 根据定义（28.10），公式（28.17）成立（它们是恒等的），所以只要证明（28.16），此可由下面得到

$$A^k = A\underline{Q}^{(k-1)}\underline{R}^{(k-1)} = \underline{Q}^{(k)}R^{(k)}\underline{R}^{(k-1)} = \underline{Q}^{(k)}\underline{R}^{(k)}.$$

其中第一个等式从（28.16）的归纳假设得到，而第二个等式由（28.8）和（28.9）得到，第三个等式从（28.15）得到.

另一方面，考虑关于 QR 算法的 $k \geqslant 1$ 情形，可以用一系列等式来证明（28.16）

$$A^k = A\underline{Q}^{(k-1)}\underline{R}^{(k-1)} = \underline{Q}^{(k-1)}A^{(k-1)}\underline{R}^{(k-1)} = \underline{Q}^{(k)}\underline{R}^{(k)}.$$

第一个等式可由（28.16）的归纳假设得到，第二个等式可由（28.17）的归纳假设得到，第三个等式可由（28.12）以及（28.14）和（28.15）得出. 最后，可以由下面序列来证明（28.17），

$$A^{(k)} = (Q^{(k)})^{\mathrm{T}} A^{(k-1)} Q^{(k)} = (\underline{Q}^{(k)})^{\mathrm{T}} A \underline{Q}^{(k)}.$$

上式中第一个等式可以从（28.12）和（28.13）得到，第二个等式可由（28.17）的归纳假设得到. □

28.5　QR 算法的收敛性

所有具体讨论已经结束了，下面将着重讨论不带位移的 QR 算法的收敛性.

首先，在定性的理解水平上：（28.16）和（28.17）是关键. 首先说明，为什么 QR 算法可以求得特征向量，它构造了逐次幂 A^k 的标准正交基. 第二个说明是为什么算法可以求出特征值. 从（28.17）可知，$A^{(k)}$ 的对角元素是对应于 $\underline{Q}^{(k)}$ 的列的 A 的瑞利商（见习题 27.1）. 当那些列收敛到特征向量时，瑞利商收敛（由（27.3），快 2 倍）到相应的特征值. 同时，（28.17）推出，$A^{(k)}$ 的非对角线元素相当于包含了在 A 的左、右两端乘的不同特征向量的近似广义瑞利商. 当这些近似收敛到不同特征向量时，它们一定变成正交，所以 $A^{(k)}$ 的非对角线元素必趋于零.

特别要强调，基本等式（28.16）和（28.17）对于理解不带位移的 QR 算法的重要性，这两个等式应记住，由此可以得到一些重要结论.

关于进一步的定性理解，有下面的定理 28.4.

217

定理28.4　设A为实对称矩阵，其特征值满足$|\lambda_1|>|\lambda_2|>\cdots>|\lambda_m|$，并且其对应的特征向量矩阵$Q$有所有非奇异的前主子矩阵，那么令纯QR算法（算法28.1）应用到A上．当$k\to\infty$时，$A^{(k)}$以常数$\max_j|\lambda_{j+1}|/|\lambda_j|$线性收敛到$\mathrm{diag}(\lambda_1,\cdots,\lambda_m)$，并且$\underline{Q}^{(k)}$（其列的符号按需要可作调整）以同样速度收敛到$Q$.

习　题

28.1　如果应用无位移的QR算法到正交矩阵将出现什么？计算出答案，然后说明与定理28.4的关系.

28.2　如果QR算法各步不能保持三对角形式，那么准备约化到三对角形式是几乎没有用的.事实并不如此.

(a) 在对称三对角矩阵A的QR因子分解$A=QR$中，R的元素一般是否非零？Q的元素是什么？（在实际中，Q的元素无显式表示.）

(b) 证明当乘积RQ形成时，三对角结构被恢复.

(c) 如何说明，吉文斯旋转或2×2豪斯霍尔德镜射算子可用到计算三对角矩阵的QR因子分解中，其计算量远比满矩阵的计算量少很多.

28.3　实对称矩阵A有重数为8的特征值1，而其他的全部特征值的绝对值$\leqslant0.1$. 试描述一个算法来求对应于优势特征值的8维特征空间的标准正交基.

28.4　考虑应用到三对角对称矩阵$A\in\mathbb{R}^{m\times m}$的算法28.1中的一步.

(a) 如果仅求特征值，那么在k步仅需要$A^{(k)}$而不需要$Q^{(k)}$，用本书中描述的标准方法，试确定从$A^{(k-1)}$得到$A^{(k)}$所需要的浮点运算量是多少.

(b) 如果要求所有的特征向量，那么矩阵$\underline{Q}^{(k)}=Q^{(1)}Q^{(2)}\cdots Q^{(k)}$也需要存储，现在试确定从$k-1$步到$k$步所需的浮点运算量是多少.

218

第 29 讲 带位移的 QR 算法

在每步引入位移 $A \to A - \mu I$，可改进 QR 迭代. 本讲将说明由于有一个与瑞利商迭代存在隐式的联系，这个思想将导致 3 次收敛.

29.1 与逆迭代的联系

仍将假定 $A \in \mathbb{R}^{m \times m}$ 是实的和对称矩阵，具有实的特征值 $\{\lambda_j\}$ 和标准正交的特征向量 $\{q_j\}$.

正如已经见过的，"纯" QR 算法（算法 28.1）等价于同时迭代应用到单位矩阵，特别地，结果的第 1 列是按幂迭代应用到 e_1 来得到的.

这个观察有一个对偶，算法 28.1 也等价于同时逆迭代（simultaneous inverse iteration）应用到"弹跳的"单位矩阵 P，特别地，结果的第 m 列是按逆迭代应用到 e_m 来得到的.

可以用下面方法来建立这一结论，如上讲一样，令 $Q^{(k)}$ 是在 QR 算法的第 k 步的正交因子，在上讲中已指出，这些矩阵的累积乘积（28.14）

$$\underline{Q}^k = \prod_{j=1}^{k} \underline{Q}^{(j)} = \left[q_1^{(k)} \middle| q_2^{(k)} \middle| \cdots \middle| q_m^{(k)} \right],$$

与出现在同时迭代的第 k 步（28.9）的正交矩阵是一样的. 另一个方法是把这个结论表示为 $\underline{Q}^{(k)}$ 是 QR 因子分解（28.16）的正交因子，

$$A^k = \underline{Q}^{(k)} \underline{R}^{(k)}.$$

下面考虑，若对上面的公式求逆，那么将出现什么情况. 计算

$$A^{-k} = (\underline{R}^{(k)})^{-1} \underline{Q}^{(k)\mathrm{T}} = \underline{Q}^{(k)} (\underline{R}^{(k)})^{-\mathrm{T}}; \qquad (29.1)$$

其中第 2 个等式用了 A^{-1} 是对称的这一事实. 令 P 为倒换行或列次序的 $m \times m$ 置换矩阵：

$$P = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \cdots & & \\ 1 & & & \end{bmatrix}.$$

由于 $P^2 = I$，（29.1）可以重写为

$$A^{-k}P = \left[\underline{Q}^{(k)}P\right]\left[P(\underline{R}^{(k)})^{-\mathrm{T}}P\right]. \tag{29.2}$$

在这个乘积中的第 1 个因子$\underline{Q}^{(k)}P$，是正交的．第 2 个因子 $P(\underline{R}^{(k)})^{-\mathrm{T}}P$，是上三角形的［从下三角形矩阵 $(\underline{R}^{(k)})^{-\mathrm{T}}$ 开始，先是倒换顶部—底部，然后再倒换左—右］．于是，(29.2) 可以解释为 $A^{-k}P$ 的 QR 因子分解．换言之，已经有效地完成了应用于初始矩阵 P 在 A^{-1} 上的同时迭代，即完成了 A 的同时逆迭代．特别地，$\underline{Q}^{(k)}P$ 的第 1 列，即 $\underline{Q}^{(k)}$ 的最后一列是应用 k 步逆迭代于向量 e_m 的结果．

29.2　与带有位移的逆迭代的联系

于是 QR 算法既是同时迭代又是同时逆迭代：完美的对称．但是，正如第 27 讲讲到的，在幂迭代和逆迭代之间有巨大的差别：逆迭代可以通过使用位移进行任意地加速．对于特征值的估计 $\mu\approx\lambda_j$ 愈好，带位移的矩阵 $A-\mu I$ 的一步逆迭代所完成的就愈多．算法 28.2 已经指明了怎样在 QR 算法的一步中引入位移，这样做恰好符合于相应的同时迭代和逆迭代过程中的位移，因此它们的实际效果是完全相同的．

令 $\mu^{(k)}$ 表示在 QR 算法的第 k 步上选择的特征值估计，由算法 28.2 知，带位移的 QR 算法的第 $k-1$ 步与第 k 步之间的关系是

$$A^{(k-1)} - \mu^{(k)}I = Q^{(k)}R(k),$$
$$A^{(k)} = R^{(k)}Q^{(k)} + \mu^{(k)}I.$$

220

这意味着

$$A^{(k)} = (Q^{(k)})^{\mathrm{T}}A^{(k-1)}Q^{(k)}, \tag{29.3}$$

用归纳法可得

$$A^{(k)} = (\underline{Q}^{(k)})^{\mathrm{T}}A\underline{Q}^{(k)},$$

此式与 (28.11) 一样，然而，(28.16) 不再成立，而有因子分解

$$(A-\mu^{(k)}I)(A-\mu^{(k-1)}I)\cdots(A-\mu^{(1)}I) = \underline{Q}^{(k)}\underline{R}^{(k)}, \tag{29.4}$$

这是关于同时迭代的带位移的变化（其证明省略）．总之，$\underline{Q}^{(k)} = \prod_{j=1}^{k}Q^{(j)}$ 是 $\prod_{j=k}^{1}(A-\mu^{(j)}I)$ 的正交化．$\underline{Q}^{(k)}$ 的第 1 列是将位移为 $\mu^{(j)}$ 的带位移的幂迭代应用于 e_1 的结果，而最后一列是将具有同样位移的带位移的逆迭代对 e_m 应用 k 步的结果．如果位移是好的特征值估计，那么$\underline{Q}^{(k)}$ 的最后一列将很快收敛到一个特征向量．

29.3　与瑞利商迭代的联系

我们已经发现，隐含于带位移的 QR 算法中的有力工具：带位移的逆迭代，为完

成这一想法，就需要一个选择位移的方法，使其在$\underline{Q}^{(k)}$的最后一列达到快速收敛.

瑞利商是好的起点. 为估计对应于$\underline{Q}^{k}$ 的最后一列近似特征向量的特征值，很自然地将瑞利商应用到最后一列，这给出了

$$\mu^{(k)} = \frac{(q_m^{(k)})^{\mathrm{T}} A q_m^{(k)}}{(q_m^{(k)})^{\mathrm{T}} q_m^{(k)}} = (q_m^{(k)})^{\mathrm{T}} A q_m^{(k)}. \tag{29.5}$$

如果这个数被选为每步的位移，那么特征值和特征向量的估计$\mu^{(k)}$和$q_m^{(k)}$ 恒等于以e_m为初值用瑞利商迭代计算得到的结果. 因此，QR 算法有 3 次收敛，其含义是$q_m^{(k)}$3 次收敛到一个特征向量.

注意到，在 QR 算法中，瑞利商$r(q_m^{(k)})$是作为$A^{(k)}$的m，m元素出现的，因此不用计算就可以得到. 在上一讲的结尾已提到过这些，但为强调起见，在此给出一个显式的推导. 以（29.3）为开始，有

$$A_{mm}^{(k)} = e_m^{\mathrm{T}} A^{(k)} e_m = e_m^{\mathrm{T}} \underline{Q}^{(k)\mathrm{T}} A \underline{Q}^{(k)} e_m = q_m^{(k)\mathrm{T}} A q_m^{(k)}. \tag{29.6}$$

因此，若简单地令$\mu^{(k)} = A_{mm}^{(k)}$，那么（29.5）与上式一样，这被称为瑞利商位移 221 (Rayleigh quotient shift).

29.4 Wilkinson 位移

虽然瑞利商位移在一般情况下给出了 3 次收敛，但是，不是对所有的初始条件都能保证收敛. 用一个简单例子来说明这一结论，考虑矩阵

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{29.7}$$

不带位移的 QR 算法根本不收敛：

$$A = Q^{(1)} R^{(1)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$A^{(1)} = R^{(1)} Q^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = A.$$

然而，由于$A_{mm}=0$，所以瑞利商位移$\mu = A_{mm}$，就没有效果. 于是，很清楚地看出，在这最坏的情况下，带有瑞利商位移的 QR 算法可能失败.

由于特征值的对称性，所以问题出现了. 一个特征值是$+1$，另一个特征值是-1，所以如果对特征值估计为 0 时，企图作出改进，有利于每个特征值的可能性是相等的，且估计是不能改进的，真正需要的是能够破坏这种对称性的特征值估计，这样的一个选择定义如下：令B表示$A^{(k)}$的右下角2×2子矩阵：

$$B = \begin{bmatrix} a_{m-1} & b_{m-1} \\ b_{m-1} & a_m \end{bmatrix}.$$

Wilkinson 位移（Wilkinson shift）定义为较靠近 a_m 的 B 的一个特征值，如果遇到复杂情况，可以任意选择 B 的两个特征值中的一个，关于 Wilkinson 位移的一个数值稳定的公式是

$$\mu = a_m - \operatorname{sign}(\delta) b_{m-1}^2 / \left(|\delta| + \sqrt{\delta^2 + b_{m-1}^2} \right), \tag{29.8}$$

其中 $\delta = (a_{m-1} - a_m)/2$. 如果 $\delta = 0$，那么，$\operatorname{sign}(\delta)$ 可以任意取为 1 或 -1.

如瑞利商位移一样，Wilkinson 位移在一般情况也可以达到 3 次收敛. 而且可以指出，在最坏的情况下，Wilkinson 位移至少可以达到 2 次收敛，特别地，带有 Wilkinson 位移的 QR 算法总是收敛的（在精确的算术运算中）.

在例（29.7）中，Wilkinson 位移不是 1，就是 -1，于是对称性被破坏了，并且一步就可得到收敛. 222

29.5　稳定性和精确度

现在已经完成了 QR 算法技巧的讨论，当然还有许多实际细节，都省略了，例如压缩条件和关于位移的“隐式”设计. 留下来的是关于稳定性和精确度的讨论.

由于使用了正交矩阵，所以可以期望 QR 算法是向后稳定的. 如前几讲所述，形成这一结果的最简单方法是，令 $\tilde{\Lambda}$ 表示用浮点运算计算的 A 的对角化，而 $\tilde{Q}$ 则表示完全正交的矩阵，它与实施方法中数值计算得到的所有豪斯霍尔德镜射（或吉文斯旋转）的乘积有关，可以证明如下定理.

定理 29.1　设实的、对称的和三对角的矩阵 $A \in \mathbb{R}^{m\times m}$，是在满足（13.5）和（13.7）的计算机上用 QR 算法（算法 28.2）来对角化的，并设 $\tilde{\Lambda}$ 和 $\tilde{Q}$ 为上面指出的矩阵，那么对于某个 $\delta A \in \mathbb{C}^{m\times m}$，有

$$\tilde{Q} \tilde{\Lambda} \tilde{Q}^* = A + \delta A, \qquad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{机器}) \tag{29.9}$$

像本书中大部分的算法一样，QR 算法得到了轻微扰动问题的精确解. 结合定量 26.1 和定理 29.1，我们看到对于计算矩阵特征值，用 QR 算法得到的三对角约化是向后稳定的算法. 为了考察计算特征值的精确度包含着的内容，可以把这一结论与涉及实对称矩阵（正规矩阵的特殊情况）特征值扰动的结果（26.4）结合起来，其结论是计算得到的特征值 $\tilde{\lambda}_j$ 满足

$$\frac{|\tilde{\lambda}_j - \lambda_j|}{\|A\|} = O(\epsilon_{机器}) \tag{29.10}$$

对于只需要 $\sim\frac{4}{3}m^3$ 次 flop 的算法不能认为是不好的，事实上，这个计算量仅相当于计算两个 $m\times m$ 矩阵乘积的$\frac{2}{3}$.

习　题

29.1 下面 5 部分问题要求仅用初等建块组建求一个实对称矩阵的全部特征值的 MATLAB 程序. 不必使用对称性或优化零结构来得到最优的常数因子. 用大约 50 行长的程序解整个问题是有可能的. 223

(a) 写一个用正交相似变换约化实对称 $m\times m$ 矩阵到三对角型矩阵的函数 `T = tridiag(A)`. 程序应该仅用初等 MATLAB 运算，例如，不能用内置函数 `hess`. 输出矩阵 `T` 在舍入误差内，应该是对称的和三对角的. 如果喜欢，加一行使 `T` 最终是精确的对称和三对角的. 考虑一个例子，把程序应用到 `A = hilb(4)`.

(b) 写出运行实三对角矩阵 T 不带位移的 QR 算法的函数 `Tnew = qralg(T)`. 关于每步的 QR 因子分解，如果可采用的话，可用习题 10.2 的程序. `[W,R] = house(A)` 和 `Q = form Q(W)`，或用 MATLAB 的命令 `qr`，或为了有更大的效率，可用基于吉文斯旋转或 2×2 豪斯霍尔德. 镜射算子而不是用 $m\times m$ 运算的新编程，还有，可使每步的矩阵是对称和三对角型. 当 $m,m-1$ 元素满足 $|t_{m,m-1}|<10^{-12}$（苛刻的工业强度收敛准则!）时，程序应停止并返回到当前的三对角矩阵 `T,Tnew`. 再次应用程序到 `A = hilb(4)`

(c) 编制一个驱动程序：(i) 调用 `triding`，(ii) 调用 `qralg`，得到一个特征值，(iii) 用一个更小的矩阵调用 `qralg` 得另一个特征值，等等，直到 $\boldsymbol{A}$ 的所有的特征值都确定. 完成这些事情以使在每一 QR 迭代时，$|t_{m,m-1}|$ 的值按向量存储并且以使在结束时，程序可以画出把这些值作为 QR 因子分解数的函数的半对数曲线（这里 m 表示从 `length(A)` 步到 `length(A) -1` 步等等直到 3，最后到 2 时为收缩开始，曲线将是锯齿状的.）运行程序 `A = hilb (4)`，输出应该是特征值集合和“锯齿曲线”.

(d) 修改 `qralg`，使其在每步用 Wilkinson 位移，对于同样例子，作出新的锯齿曲线.

(e) 对于矩阵 `A = diag(15:-1:1) + ones(15,15)` 重新运行上述程序，得到对应于有位移和无位移的两条锯齿曲线. 讨论这里的矩阵和较早的矩阵的收敛速度比. 收敛是线性、超线性、2 次、3 次的还是更高次数? 谈论“每个特征值 QR 迭代数”有意义吗? 224

第30讲 其他的特征值算法

特征值的计算除了QR算法外，还有别的方法. 在本讲中仅简单地叙述关于实对特征值问题的3种著名的可用方法：满矩阵的雅可比算法，三对角矩阵的二分算法和分而治之算法.

30.1 雅可比算法

计算矩阵特征值的最古老观念之一是于1845年由雅可比引入的雅可比算法. 在整个计算机时代，尽管雅可比方法没有真正地成为主流算法，但这个方法一直引起关注，特别自从出现了并行计算以来更是如此.

这想法如下，对于阶为5或大于5的矩阵，特征值只能用迭代求得（第25讲）. 然而，比这更小的矩阵能一步处理. 为什么不把$\boldsymbol{A}$的一个小子矩阵对角化，然后对角化另一个，如此下去，最终希望收敛到这完全矩阵的一个对角化？

这个想法已经用4×4子矩阵作了试验，但标准的方法是基于2×2子矩阵. 一个2×2的实对称矩阵以如下形式来对角化，

225

$$\boldsymbol{J}^{\mathrm{T}}\begin{bmatrix} a & d \\ d & b \end{bmatrix}\boldsymbol{J}=\begin{bmatrix} \neq0 & 0 \\ 0 & \neq0 \end{bmatrix}, \tag{30.1}$$

其中$\boldsymbol{J}$是正交矩阵，选择$\boldsymbol{J}$有几种方法，可以取为如下形式的2×2豪斯霍尔德镜射算子

$$\boldsymbol{F}=\begin{bmatrix} -c & s \\ s & c \end{bmatrix}, \tag{30.2}$$

其中，对于某个θ，$s=\sin\theta$和$c=\cos\theta$，注意到，$\det F=-1$，这是镜射算子的特点. 另一选择是，可以不用镜射算子而用旋转

$$\boldsymbol{J}=\begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{30.3}$$

其中$\det\boldsymbol{J}=1$，这是雅可比算法的标准方法. 可以证明，如果θ满足

$$\tan(2\theta)=\frac{2d}{b-a}, \tag{30.4}$$

那么对角化（30.1）可以完成，基于这种选择矩阵$\boldsymbol{J}$称为雅可比旋转（它与吉文斯旋转有同样形式（习题10.4）；惟一的差别在于θ的选取使得$\boldsymbol{J}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{J}$是对角矩阵而不

是 $J^{\mathrm{T}}A$ 为三角形矩阵.).

现设 $A\in\mathbb{R}^{m\times m}$ 是对称矩阵，雅可比算法是由重复应用基于由（30.3）和（30.4）定义的矩阵的变换（30.1）所组成. 将矩阵 J 扩大到 $m\times m$ 矩阵，此矩阵除了具有形式（30.3）的 4 个元素外为单位矩阵. 在 A 的左边应用 J^{T} 将修改 A 的两行，在 A 的右边应用 J 将修改 A 的两列. 在每步，在矩阵中引入了对称的一对零，但以前的零将被破坏. 然而，正如 QR 算法一样，一般的效果是不断地减少这些非零元素的值.

在每步上哪些非对角线元素 a_{ij} 将化为零? 当然适合于计算的方法是在每步挑选出最大的非对角线元素. 由于可以证明，在每步非对角元素的平方和至少以因子 $1-2/m^2-m$ 减少（见习题 30.3），所以收敛性分析就变成很容易的事了. 经 $O(m^2)$ 步后，其中每步需要的运算量为 $O(m)$，平方和必须以常数因子下降，因此收敛到精确度 $\epsilon_{机器}$ 将经过 $O(m^3\log(\epsilon_{机器}))$ 运算后得到. 事实上，我们知道收敛速度比这更好，最后是平方收敛而不是线性收敛，所以实际运算计数为 $O(m^3\log(|\log(\epsilon_{机器})|))$（习题 25.2).

在计算机上，为避免用 $O(m^2)$ 次运算来寻找最大的元素，所以一般用循环方式来消除非对角线元素. 例如，如果对角线以上 $m(m-1)/2$ 个元素以 $a_{12},a_{13},\cdots$ 为开始，用最简单的按行的方式次序来消元，那么快的渐近收敛就再次得到保证. 在 2×2 运算（包含了所有 $m\times(m-1)/2$ 对非对角元素）的一次搜索（sweep）后，精确度一般是以好于一个常数因子来改进的，而且，收敛最后达到二阶. 226

因为雅可比方法仅同时处理行和列的成对元素，使其容易并行化（习题 30.4），因而它是一个有吸引力的方法. 矩阵预先不做三对角化，雅可比旋转将破坏三对角化结构. 维数 $m\leqslant1000$ 的矩阵的收敛一般比 10 次搜索要少得多，并且最后的分量方式的精确度一般比 QR 算法得到的更好一些. 可惜的是，甚至在并行机上，雅可比算法也不总是比三对角化之后的 QR 算法或分而治之算法（下面讨论）的快（习题 30.2)，但通常在 10 的因子范围内.

30.2 二 分 法

另一个特征值算法是在实际中非常重要的一个算法，称为*二分法*（bisection). 对称矩阵三对角化后，如果不要求求出全部特征值而仅需求出部分特征值，那么二分法是标准的下一步方法. 例如，二分法可以求出最大的 10% 的特征值，或最小的 30% 的特征值，或在区间［1.2］内的所有特征值. 一旦求得希望的特征值，那么对应的特征向量可以由一步逆迭代求得（算法 27.2).

二分法的出发点是初等的. 由于实对称矩阵的特征值是实的，因此可以用在实线上进行搜索求出多项式 $p(x)=\det(A-xI)$ 根的方法来找出 A 的特征值. 这听起来好像是个坏的想法，因为在第 15 讲和第 25 讲中已经说过，为求特征

值而采用多项式求根是极其不稳定的方法. 然而两者是有区别的，其差别是那些评论适合于从多项式系数求根的想法. 现在，其想法是用在不同点 x 上来计算 $p(x)$ 的值，而不考虑其系数，并用通常的关于非线性函数的二分过程来求多项式的根. 例如，用选主元的高斯消元法可以做到这一点（习题21.1），并且得到的算法是高度稳定的.

这些似乎非常有用，但不是很令人振奋. 二分法所表现的能力和感染力是特征值和行列式的若干附加性质——这些性质不很明显.

给定一个对称矩阵 $A\in\mathbb{R}^{m\times m}$，令 $A^{(1)},\cdots,A^{(m)}$ 表示其阶数为 $1,\cdots,m$ 的主（即左上）方子矩阵. 可以证明，这些矩阵的特征值是*交织的*（interlace）. 在定义这个性质之前，首先假定 A 是三对角的且是*不可约的*，其意义为所有非对角元素不等于零，其中 A 为

$$A=\begin{bmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & b_2 & a_3 & \ddots & \\ & & \ddots & \ddots & b_{m-1} \\ & & & b_{m-1} & a_m \end{bmatrix}, b_j\neq 0. \tag{30.5}$$

227

（如果在非对角线上有零元素，那么特征值问题能够被压缩，如在算法28.2中）由习题25.1，$A^{(k)}$ 的特征值是不同的，将它们表示为 $\lambda_1^{(k)}<\lambda_2^{(k)}<\cdots<\lambda_k^{(k)}$. 使二分法有效的关键性质是，这些特征值是*严格交织的*（strictly interlace），即对于 $k=1,2,\cdots,m-1$ 和 $j=1,2,\cdots,k-1$，满足不等式

$$\lambda_j^{(k+1)}<\lambda_j^{(k)}<\lambda_{j+1}^{(k+1)}. \tag{30.6}$$

这个性质的示意图见图30-1.

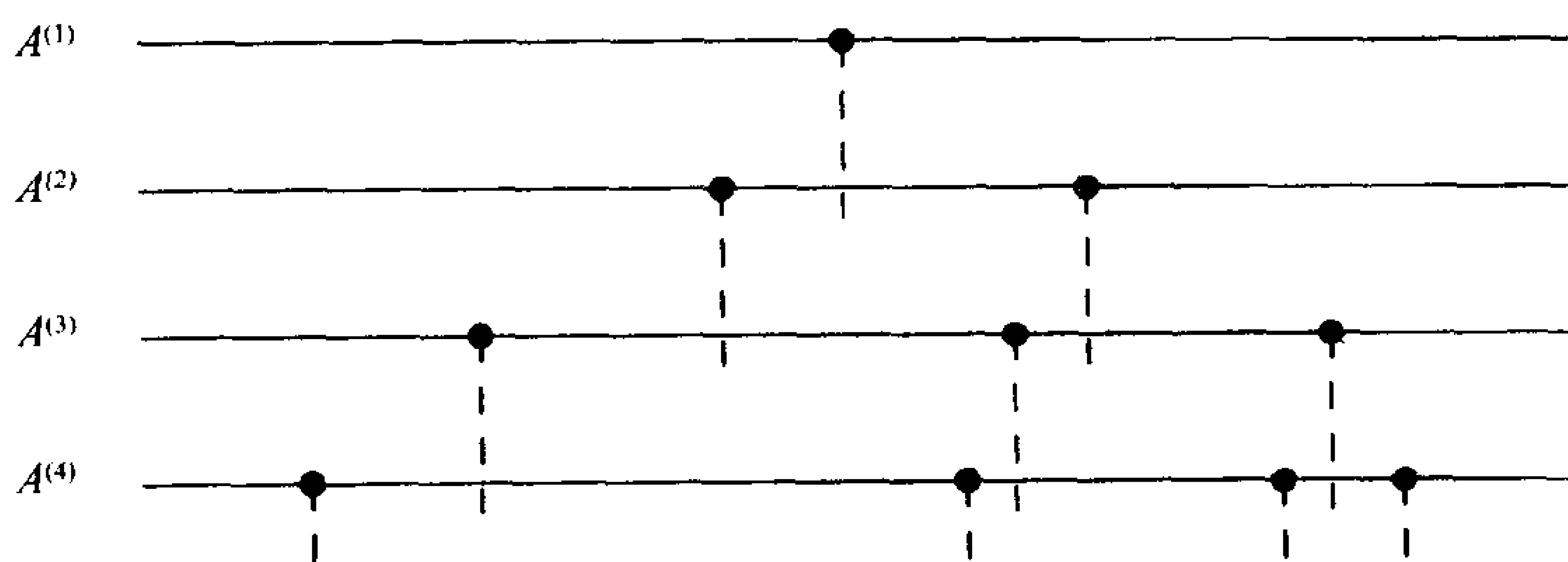


图30-1　对不可约三对角实对称矩阵 A 的主子矩阵 $\{A^{(j)}\}$ 的严格特征值交织性质（30.6）的说明. $A^{(k)}$ 的特征值交织 $A^{(k+1)}$ 的特征值. 二分算法采用了此性质的优点

使用交织性质就有可能计算矩阵在指定区间内特征值的个数. 例如，考虑4×4三对角矩阵

$$A = \begin{bmatrix} 1 & 1 & & \\ 1 & 0 & 1 & \\ & 1 & 2 & 1 \\ & & 1 & -1 \end{bmatrix}.$$

从下面的数

$$\det(A^{(1)}) = 1, \det(A^{(2)}) = -1, \det(A^{(3)}) = -3, \det(A^{(4)}) = 4,$$

可以知道，$A^{(1)}$没有负特征值，$A^{(2)}$有一个负特征值，$A^{(3)}$有一个负特征值以及$A^{(4)}$有两个负特征值，一般地，对任意的对称三对角矩阵$A \in \mathbb{R}^{m\times m}$，负特征值个数等于下面序列中符号的改变的次数，

$$1, \det(A^{(1)}), \det(A^{(2)}), \cdots, \det(A^{(m)}). \tag{30.7}$$

此序列称为 Sturm 序列（Sturm sequence）.（如果定义“符号改变”为从十或零到一，或从一或零到十，但不能是从十或一到零的一个转变，那么即使遇到行列式为零时，此规定仍可以使用.）用单位矩阵的倍数来作 A 的位移，以此可以确定在任何区间$[a,b)$内特征值的数目：它是在（$-\infty, b$）内特征值的数目减去在（$-\infty, a$）内特征值的数目. 228

再观察一次就可以完成二分法的叙述：对于三对角矩阵，矩阵 $\{A^{(k)}\}$ 的行列式可以用三项递推关系来联系. 从（30.5），将 det（$A^{(k)}$）展成第k行元素b_{k-1}和a_k与它自己的代数余子式的乘积之和

$$\det(A^{(k)}) = a_k \det(A^{(k-1)}) - b_{k-1}^2 \det(A^{(k-2)}). \tag{30.8}$$

用 xI 引入位移，记$p^{(k)}(x) = \det(A^{(k)} - xI)$，可以得到

$$p^{(k)}(x) = (a_k - x)p^{(k-1)}(x) - b_{k-1}^2 p^{(k-2)}(x). \tag{30.9}$$

如果定义$p^{(-1)}(x)=0$ 和$p^{(0)}(x)=1$，那么这个递推式对全部$k=1,2,\cdots,m$都成立.

对于一系列x值应用（30.9），并以此来计录符号的改变，这样二分算法确定了在任何小的区间内的特征值. 对于此序列的每个 值的计算代价为$O(m)$次 flop，因此求一个相对精度为$\epsilon_{机器}$的特征值，总的代价为$O(m \log(\epsilon_{机器}))$次 flop. 如果需要计算一个小的特征值，那么此算法比 QR 算法的$O(m^2)$运算计数有明显的改进. 在多处理系统的计算机上，多重特征值可以在单独的处理机上独立地求出.

30.3 分而治之

分而治之算法表示了 20 世纪 60 年代以来在矩阵特征值计算中最重要的进展，它基于把对称三对角特征值问题递归细分到更小阶数的问题. 首先由 Cuppen 在 1981 年引进，如果需要计算特征向量和特征值，那么分而治之算法比 QR 算法要快 2 倍多.

下面将仅给出基本思想，而省略全部细节. 但是要警告读者，在这个算法中，细节是特别重要的，其原因是该算法不是完全稳定的，除非它们被真正理解——问题是Cuppen的原始论文发表后10年间事情仍未弄清楚.

设$\boldsymbol{T}\in\mathbb{R}^{m\times m}$，$m\geqslant 2$，是对称的，三对角的以及在非对角仅有非零的意义下的不可约矩阵（否则，问题是可以压缩的.）那么对任意的$n,1\leqslant n<m,\boldsymbol{T}$可以分解为如下的子矩阵：

229

$$\boldsymbol{T}=\left[\begin{array}{c|c}\boldsymbol{T}_1 & \beta \\ \hline \beta & \boldsymbol{T}_2\end{array}\right]=\left[\begin{array}{c|c}\hat{\boldsymbol{T}}_1 & \\ \hline & \hat{\boldsymbol{T}}_2\end{array}\right]+\left[\begin{array}{c|c}\beta & \beta \\ \hline \beta & \beta\end{array}\right] \tag{30.10}$$

其中，$\boldsymbol{T}_1$为$\boldsymbol{T}$的左上$n\times n$主子矩阵，$\boldsymbol{T}_2$为$\boldsymbol{T}$的右下$(m-n)\times(m-n)$主子矩阵，$\beta=t_{n+1,n}=t_{n,n+1}\neq 0$，$\boldsymbol{T}_1$和$\hat{\boldsymbol{T}}_1$的差别仅为右下的元素$t_{nn}$被$t_{nn}-\beta$来代替，$\boldsymbol{T}_2$和$\hat{\boldsymbol{T}}_2$的差别仅为左上的元素$t_{n+1,n+1}$被$t_{n+1,n+1}-\beta$来代替. 引入了两个元素的修改使得（30.10）最右边矩阵的秩为1.

下面为用语言来表示（30.10）：**三对角矩阵可以写为有三对角块的2×2块对角矩阵与秩1的校正矩阵之和.**

分而治之算法如下进行，以$n\approx m/2$把矩阵$\boldsymbol{T}$分解为（30.10），假定$\hat{\boldsymbol{T}}_1$和$\hat{\boldsymbol{T}}_2$的特征值已知，由于校正矩阵的秩为1，所以用非线性但快速计算来由$\hat{\boldsymbol{T}}_1$和$\hat{\boldsymbol{T}}_2$的特征值得到$\boldsymbol{T}$本身的特征值. 重复这想法，用进一步带有秩1校正的细分可求出$\hat{\boldsymbol{T}}_1$和$\hat{\boldsymbol{T}}_2$的特征值，等等. 照这样，$m\times m$特征值问题可约化为一组1×1特征值问题和秩1校正的族.（在实际中，为了最值效率，当子矩阵的维数充分小，习惯切换到QR算法，而不是继续迭代.）

在这个过程中，有一个关键的数学点. 如果$\hat{\boldsymbol{T}}_1$和$\hat{\boldsymbol{T}}_2$的特征值是已知的，那么怎样求出$\boldsymbol{T}$的特征值？为回答这个问题，假定对角化

$$\hat{\boldsymbol{T}}_1=\boldsymbol{Q}_1\boldsymbol{D}_1\boldsymbol{Q}_1^{\mathrm{T}},\quad \hat{\boldsymbol{T}}_2=\boldsymbol{Q}_2\boldsymbol{D}_2\boldsymbol{Q}_2^{\mathrm{T}}$$

已经计算出来，那么从（30.10）可以得出

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{Q}_1 & \\ & \boldsymbol{Q}_2\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{D}_1 & \\ & \boldsymbol{D}_2\end{bmatrix}+\beta\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}\right)\begin{bmatrix}\boldsymbol{Q}_1^{\mathrm{T}} & \\ & \boldsymbol{Q}_2^{\mathrm{T}}\end{bmatrix} \tag{30.11}$$

其中$\boldsymbol{z}^{\mathrm{T}}=(\boldsymbol{q}_1^{\mathrm{T}},\boldsymbol{q}_2^{\mathrm{T}})$，$\boldsymbol{q}_1^{\mathrm{T}}$是$\boldsymbol{Q}_1$的最后一行，$\boldsymbol{q}_2^{\mathrm{T}}$是$\boldsymbol{Q}_2$的第一行. 由于这个等式是相似变换，所以已经把这个数学问题约化为求一个对角矩阵加上秩1校正的特征值问题.

230

为说明如何求解，化简符号如下. 假定要求$\boldsymbol{D}+\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}$的特征值，其中$\boldsymbol{D}\in\mathbb{R}^{m\times m}$为具有不同对角线元素［$d_j$］的对角矩阵，$\boldsymbol{w}\in\mathbb{R}^m$是向量.（上面正号的选取对应于

$\beta>0$；对于$\beta<0$，将考虑$\boldsymbol{D}-\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}$）可以假定对于全部$j$有$w_j\neq0$，否则，问题是可约的. 那么$\boldsymbol{D}+\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}$的特征值是有理函数

$$f(\lambda)=1+\sum_{j=1}^{m}\frac{w_j^2}{d_j-\lambda}\tag{30.12}$$

的根，如图 30-2 的图解. 这一结论可如下证明，如果对某个$\boldsymbol{q}\neq\boldsymbol{0}$，$(\boldsymbol{D}+\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}})\boldsymbol{q}=\lambda\boldsymbol{q}$，那么有$(\boldsymbol{D}-\lambda\boldsymbol{I})\boldsymbol{q}+\boldsymbol{w}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{q})=\boldsymbol{0}$，由此推出$\boldsymbol{q}+(\boldsymbol{D}-\lambda\boldsymbol{I})^{-1}\boldsymbol{w}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{q})=\boldsymbol{0}$，即是，$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{q}+\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{D}-\lambda\boldsymbol{I})^{-1}\boldsymbol{w}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{q})=\boldsymbol{0}$. 这相当于方程$f(\lambda)(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{q})=\boldsymbol{0}$，其中$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{q}$必须非零，否则，$\boldsymbol{q}$为$\boldsymbol{D}$的特征向量，因此仅在一个位置上非零，最后得出$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{q}\neq\boldsymbol{0}$. 由此推出，如果$\boldsymbol{q}$是$\boldsymbol{D}+\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}$属于特征值$\lambda$的特征向量，那么$f(\lambda)$必为零，其逆则由$f(\lambda)$的形式保证了$f(\lambda)$恰有$m$个零点而得到. 方程$f(\lambda)=0$称为**特征方程**.

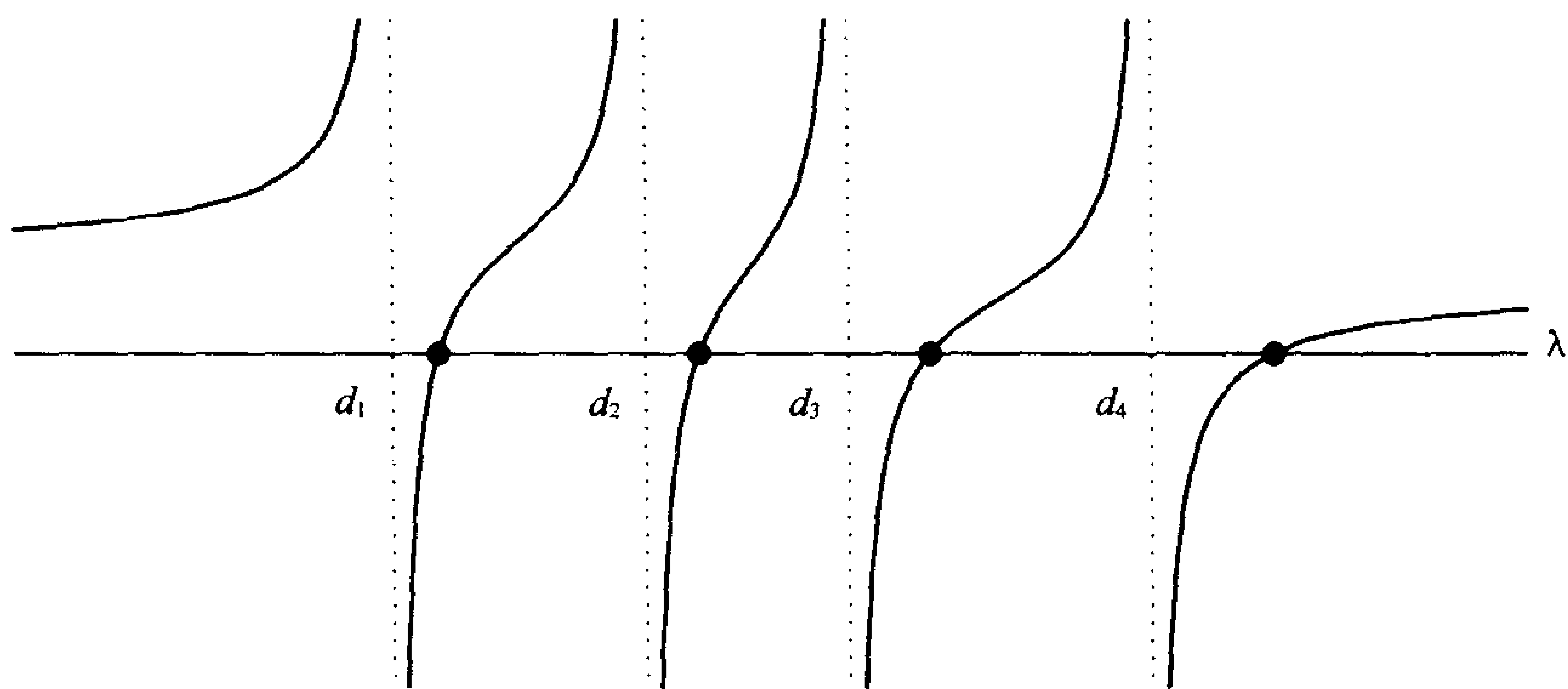


图 30-2　对于 4 阶问题的函数（30.12）$f(\lambda)$的图形，$f(\lambda)$的极点是$\boldsymbol{D}$的特征值$\{d_j\}$，$f(\lambda)$的根（实点）是$\boldsymbol{D}+\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}$的特征值. 这些根的快速确定是根据分而治之算法的每一递归步来得出的

在分而治之算法的每一递归步上，(30.12) 的根可以用一个与 Newton 方法有关的快速迭代方法求得. 对于每个根仅需$O(1)$次迭代（或若把$\epsilon_{机器}$作为变量，则为$O(\log)(|\log(\epsilon_{机器})|)$）次迭代，而对于$m\times m$矩阵，每个根的运算计数为$O(m)$次 flop，或总的运算计数为$O(m^2)$次 flop. 若设想在每步把$m$阶矩阵恰好分解为两半的递归，那么由分而治之算法求三对角矩阵的特征值的总运算计数为 231

$$O\left(m^2+2\left(\frac{m}{2}\right)^2+4\left(\frac{m}{4}\right)^2+8\left(\frac{m}{8}\right)^2+\cdots+m\left(\frac{m}{m}\right)^2\right),\tag{30.13}$$

由于在分母中的平方，所以级数收敛到$O(m^2)$（不是$O(m^2\log m)$）. 于是，这里的运算计数与 QR 算法具有同样的阶$O(m^2)$.

到目前为止，还不清楚为什么分而治之算法是有利的. 由于一个满矩阵约化到三对角型（"阶段 1"，依据第 25 讲中的术语）需要$4m^3/3$次 flop（26.2），看来，对得到

的三对角矩阵进行对角化（“阶段2”）所需$O(m^2)$次运算计数的任何改进是不太重要了. 然而，如果要求同时计算特征向量和特征值，那么，分而治之法就比较经济了. 于是，阶段1需要$8m^3/3$次flop，但阶段2也需要$O(m^3)$次flop——对于QR算法大约为$6m^3$. 由于分而治之算法的非线性迭代仅包含标量函数（30.12），不包含正交矩阵Q_j，而QR算法的每一步必须利用矩阵Q_j，所以分而治之算法减少了运算数.

运算计数显示如下，分而治之计算的$O(m^3)$部分是在（30.11）中乘以Q_j和Q_j^{T}. 在递归的所有步上的运算计数相加得到总的运算计数，其值为$4m^3/3$次flop，这比起大约$6m^3$次flop是巨大的改进. 加上阶段1的$8m^3/3$次flop给出了由大约$9m^3$到$4m^3$的改进.

实际上，由于非初等的原因，分而治之算法通常进行得甚至比这更好，对于大多数矩阵A，在（30.11）中出现的很多向量z和矩阵Q_j是数值上稀疏的，所谓数值上稀疏是指，很多元素的相对量小于机器精度. 稀疏性允许数值压缩（numerical deflation）过程，因此逐个的三对角特征值问题约化为更小维数的非耦合问题. 在典型情况下，这把阶段2的运算计数减少到比m^3次flop少一个数量级，把阶段1和阶段2的运算计数总和减少到$8m^3/3$. 对于仅求特征值来说，（30.13）变成一个过高的估计，阶段2的运算计数减少到比m^2次flop低一个数量级. 压缩的这一惊奇现象的本质在于，大多数三对角矩阵的多数特征值是“指数规律局部化的”（习题30.7）该事实是如此的显然，就如同物理学家认为玻璃是透明的一样. 在此不作进一步讨论.

在以上讨论中，似乎只有单一分而治之的算法，但事实上，有许多变形. 出于考虑，稳定性的原因，经常使用更为复杂的秩1修正，有时也用秩2修正. 求$f(\lambda)$的根有各种方法，对于大的m，实现乘以Q_j的最快方法是，用多极展开而不是用显而易见的算法. 分而治之算法的高质量实施可以在LAPACK库中找到. 232

习　题

30.1　推导公式（30.4），并给出变换（30.1）（基于θ的这个选择）的准确的几何解释.

30.2　对于雅可比算法的一步（30.1）需要多少次flop? 对于$m(m-1)/2$这么多步，即一次搜索，需要多少次flop? 一次搜索的运算计数与实对称矩阵的三对角化，以及用QR算法求其特征值的总运算计数作比较，情况如何?

30.3　证明如果在雅可比算法的每步消去了最大的非对角线元素，那么在每步上非对角线元素平方之和至少以因子$1-2/(m^2-m)$减少.

30.4　设m为偶数，计算机上有$m/2$个处理器，如果$m/2$个变换（30.1）包含了不相交的行/列对$(1,2),(3,4),(5,6),\cdots,(m-1,m)$，试说明如何将$m/2$个变换能并行地实行.

30.5　写出一个用标准行-方式次序的雅可比算法求$m\times m$实对称矩阵特征值的程序. 画出在对数尺度下，非对角线元素平方之和作为搜索数的函数，对于阶为20,40和80的随机矩阵应用此程序.

30.6 设

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 3 \end{bmatrix},$$

$\boldsymbol{A}$ 在区间 $[1,2]$ 内有多少个特征值？用二分法，使用递推式（30.9），在纸上算出其答案.

30.7 构造一个 100 阶的随机实对称三对角矩阵 $\boldsymbol{T}$，并计算其特征值分解，$\boldsymbol{T}=\boldsymbol{QDQ}^{\mathrm{T}}$. 在对数尺度下，画出几个特征向量（$\boldsymbol{Q}$ 的几个列的绝对值），并观察其局部化现象. $\boldsymbol{Q}$ 的 10 000个元素在量级上大于 10^{-10}的比例是什么？如果不用随机矩阵，而取 $\boldsymbol{T}$ 为元素为 $1,-2,1$ 的离散拉普拉斯矩阵，那么答案是什么？

233

第31讲　计　算　SVD

任意矩阵的SVD的计算能被约化到计算埃米尔特方阵的特征值分解上，但是这样做的最显然使用的方法是不稳定的．的确，计算SVD的标准方法是隐式地基于另外一种类型的约化——约化到埃米尔特形式．考虑到运算速度，首先把矩阵酉双对角化．

31.1　A 的SVD和 A^*A 的特征值

如在定理5.4中所叙述的，$m\times n$ 矩阵 A（$m\geqslant n$）的SVD，$A=U\Sigma V^*$ 是与矩阵 A^*A

$$A^*A = V\Sigma^*\Sigma V^*$$

的特征值分解有关．于是，从数学上讲，可以如下计算 A 的SVD

1. 形成 A^*A；
2. 计算特征值分解 $A^*A=V\Lambda V^*$；
3. 令 Σ 为 Λ 的 $m\times n$ 非负对角的平方根；
4. 对于酉矩阵 U，解方程组 $U\Sigma=AV$（例如，通过QR因子分解）．

这个算法经常为人们所采用，而使用者也会对SVD会有重新的认识．矩阵 A^*A 称为 A 的*协方差矩阵*（covariance matrix），它在统计和其他领域中有为人熟知的解释．然而，由于这个算法是把SVD问题约化到特征值问题，而特征值问题对扰动可能有更多的灵敏性，所以这个算法是不稳定的．

234

这个困难能够如下说明．已经注意到，当埃米尔特矩阵 A^*A 是被 δB 扰动时，每个特征值中绝对值的变化范围是由扰动的2-范数确定的．由习题26.3（b）得 $|\lambda_k(A^*A+\delta B)-\lambda_k(A^*A)|\leqslant\|\delta B\|_2$．如由下面的等式（31.2）推得，对于 A 本身的奇异值，相似的界成立，$|\sigma_k(A+\delta A)-\sigma_k(A)|\leqslant\|\delta A\|_2$．于是，计算奇异值的向后稳定算法将得到 $\tilde{\sigma}_k,\tilde{\sigma}_k$ 满足

$$\tilde{\sigma}_k=\sigma_k(A+\delta A),\quad \frac{\|\delta A\|}{\|A\|}=O(\epsilon_{机器}),\tag{31.1}$$

此式将推得

$$|\tilde{\sigma}_k-\sigma_k| = O(\epsilon_{机器}\|A\|).$$

如果用计算 $\lambda_k(A^*A)$ 来进行，观察有什么发生．如果 $\lambda_k(A^*A)$ 被稳定地计算，那么必须期望误差的阶为

$$|\tilde{\lambda}_k - \lambda_k| = O\left(\epsilon_{\text{机器}} \|A^*A\|\right) = O\left(\epsilon_{\text{机器}} \|A\|^2\right).$$

开方得到 σ_k，有

$$|\tilde{\sigma}_k - \sigma_k| = O(|\tilde{\lambda}_k - \lambda_k| / \sqrt{\lambda_k}) = O(\epsilon_{\text{机器}} \|A\|^2 / \sigma_k).$$

由于此式比前面多了因子 $O(\|A\|/\sigma_k)$，所以情况更差. 对于 A 的最优势奇异值，$\sigma_k \approx \|A\|$，这就不会有问题，但对于满足 $\sigma_k \ll \|A\|$ 的任一奇异值，这就是一个大问题. 对于最小的奇异值 σ_n，必须预料到损失 $\kappa(A)$（条件数的平方）阶的精度，这正如对某些最小二乘问题使用法方程一样（第 19 讲）.

31.2 约化到特征值问题的一个不同方法

存在一个不同的，稳定的方法把 SVD 约化到特征值问题. 假定 A 为 $m = n$ 的方阵，由于我们将看到，长方形奇异值问题可以化为方形奇异值问题，所以这一假定是非本质的限制. 考虑较早在习题 5.4 中提到过的 $2m \times 2m$ 埃米尔特矩阵

$$H = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix}. \tag{31.2}$$

因为 $A = U\Sigma V^*$ 推得 $AV = U\Sigma$ 和 $A^*U = V\Sigma^* = V\Sigma$，所以有块 2×2 方程组

$$\begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix} \begin{bmatrix} V & V \\ U & -U \end{bmatrix} = \begin{bmatrix} V & V \\ U & -U \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix} \tag{31.3}$$

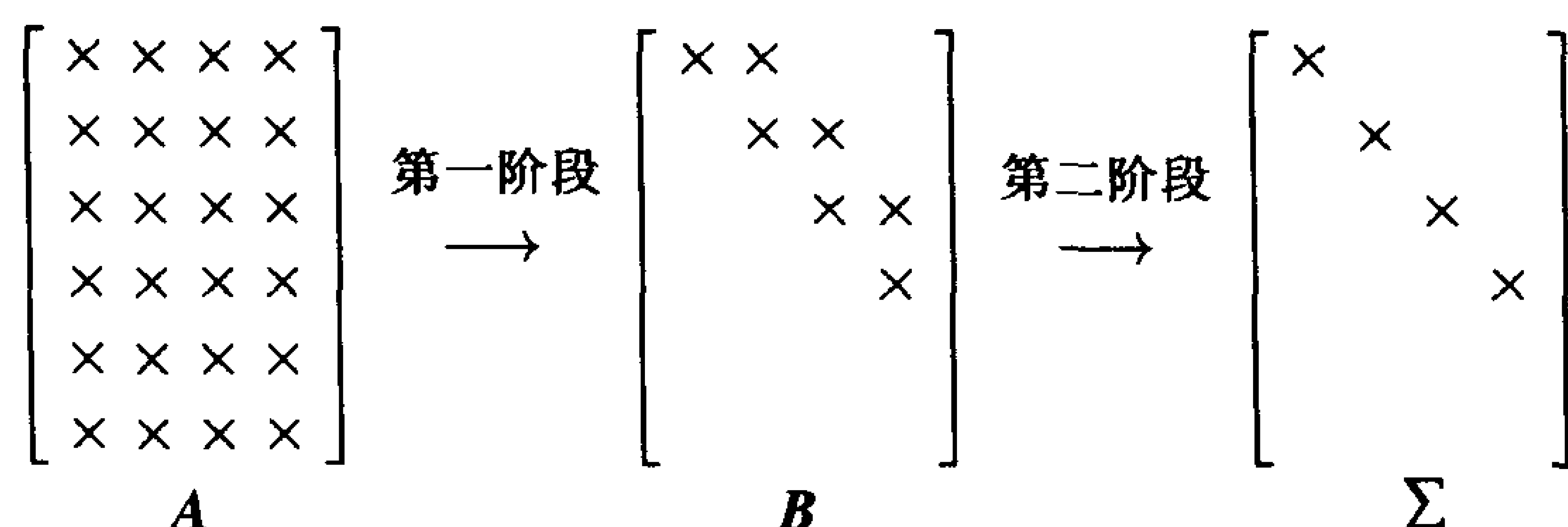
235

这相当于 H 的特征值分解. 于是可以看到，A 的奇异值是 H 的特征值的绝对值，A 的奇异向量可以由 H 的特征向量开方得到.

于是，方阵 A 可形成 H 并计算出其特征值分解，由此得到 A 的 SVD. 相对于利用 AA^* 或 A^*A，这个方法是稳定的. SVD 的标准算法都基于这个思想，即使在隐式方法中没有明显地形成 $m+n$ 那样大阶的矩阵，其基本思想不变. 快速实施这一过程的关键步骤是把初始酉约化到双对角型.

31.3 二个阶段

已经看到，埃米尔特特征值问题通常是用二个阶段的计算来求解的. 首先把矩阵约化为三对角矩阵，然后把三对角矩阵对角化，由于 Golub, Kahan 和其他人在 20 世纪 60 年代所作的工作，对于 SVD 类似于二阶段的方法已经是标准的方法了. 矩阵化为双对角形式，然后把双对角矩阵对角化：



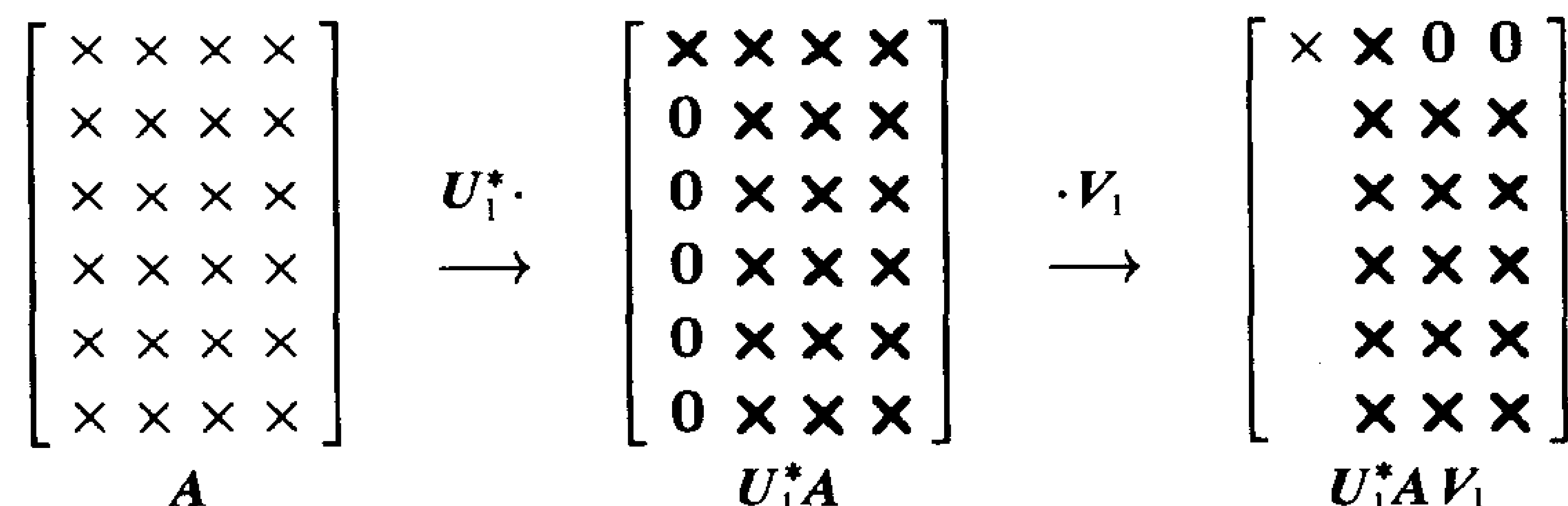
阶段1包含有限次运算，$O(mn^2)$次 flop. 阶段2在原则上需要无限多次运算，但是，标准算法超线性收敛，于是对于收敛到阶$\epsilon_{机器}$，仅需要 $O(n\log(|\log(\epsilon_{机器})|))$次迭代（习题25.2）. 实际上，把$\epsilon_{机器}$作为常数，因此 $O(n)$次迭代就达到收敛. 由于运算的矩阵是双对角线矩阵，每次这样的迭代仅需 $O(n)$次 flop. 因此阶段2总共需要 $O(n^2)$次 flop（假设仅要求奇异值而不需要求奇异向量）. 于是，即使阶段1是有限的而阶段2在原则上是无限的，但在实际中后者的消耗少得多，这正如在对称特征值问题中所看到的一样.

31.4　Golub-Kahan 双对角化

在 SVD 计算的阶段1中，在 A 的左边和右边也应用不同的酉运算来使 A 变成双对角线型. 注意这与特征值计算是不同的，在特征值计算中必须以同样的酉运算运用到两边，所以每步是相似变换. 在那种情况中，只有可能把零引入到第一次对角线的下面. 而此处，能完全三角形化，且也能把零引入到第一上对角线的上面.

236

完成这些，最简单方法是 Golub-Kahan 双对角化（Golub-Kahan bidiagonalization），进行如下. 豪斯霍尔德镜射算子交替地应用在左边和右边，每一左镜射在对角线下引入一列零，紧接着右镜射在第一上对角线的右边引入一行零，这不改变刚引入的列上的零. 例如，对于一个 6×4 矩阵，最初两对镜射算子如下：



$$\xrightarrow{U_2^* \cdot} \begin{bmatrix} \times & \times & & \\ & \mathbf{X} & \mathbf{X} & \mathbf{X} \\ & 0 & \mathbf{X} & \mathbf{X} \\ & 0 & \mathbf{X} & \mathbf{X} \\ & 0 & \mathbf{X} & \mathbf{X} \\ & 0 & \mathbf{X} & \mathbf{X} \end{bmatrix} \xrightarrow{\cdot V_2} \begin{bmatrix} \times & \times & & \\ & \times & \mathbf{X} & 0 \\ & & \mathbf{X} & \mathbf{X} \\ & & \mathbf{X} & \mathbf{X} \\ & & \mathbf{X} & \mathbf{X} \\ & & \mathbf{X} & \mathbf{X} \end{bmatrix}.$$
$$\qquad\qquad U_2^* U_1^* A V_1 \qquad\qquad\qquad U_2^* U_1^* A V_1 V_2$$

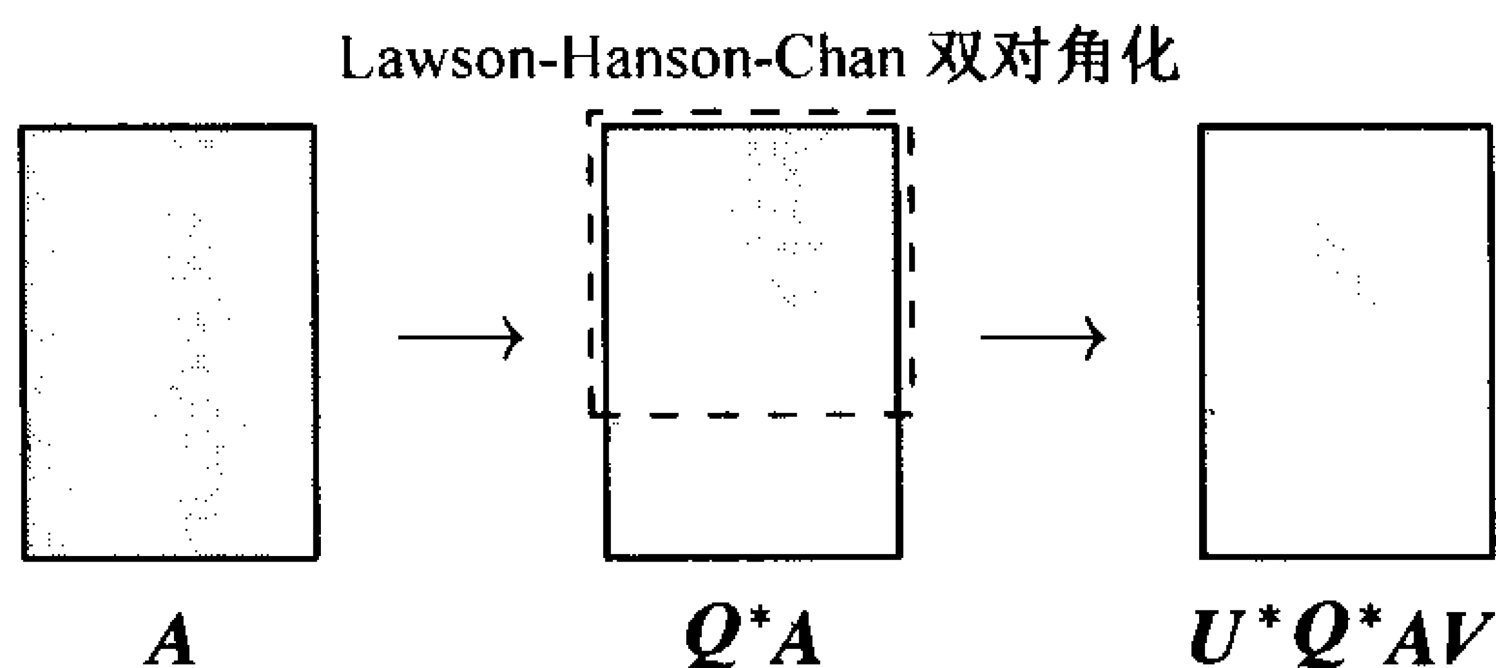
用 U_1^* 左乘修改了第 1 行至第 m 行，并在位于对角线下的第 1 列上引入零. 用 V_1 右乘修改了第 2 列至第 n 列，并在第 1 行引入零同时没有破坏第 1 列中的零. 这过程可以继续进行从第 2 行到第 m 行的运算，然后从第 3 列到第 n 列的运算，等等.

在此过程的结尾，n 个镜射算子已经应用到左边，$n-2$ 个镜射算子应用到右边. 浮点运算的模式类似于两个豪斯霍尔德 QR 因子分解相互交替，一个作用于 $m\times n$ 矩阵 $\boldsymbol{A}$，另一个作用于 $n\times m$ 矩阵 $\boldsymbol{A}^*$. 因此，总的运算计数 2 倍于 QR 因子分解的运算计数（10.9），即是，

$$\text{Golub-Kahan 双对角化的工作量：大约为 } 4mn^2-\frac{4}{3}n^3 \text{ 次 flop.} \qquad (31.4)$$

31.5 阶段 1 的更快的方法

对于 $m\gg n$，运算次数不必很大. 单个 QR 因子分解应在对角线下面都引入零，对于 $m\gg n$，有所需的大量零. 可是，Golub-Kahan 方法的运算次数高了 1 倍. 这个观察启发了对于 $m\gg n$ 双对角化的另一方法，此方法首先由 Lawson 和 Hanson 提出，后来由 Chan 发展 LHC 双对角化（LHC bidiagonalization）的思想说明如下： 237



以计算 QR 因子分解 $\boldsymbol{A}=\boldsymbol{QR}$ 开始，然后计算 $\boldsymbol{R}$ 的 Golub-Kahan 双对角化 $\boldsymbol{B}=\boldsymbol{U}^*\boldsymbol{RV}$. QR 因子分解需 $2mn^2-\frac{2}{3}n^3$ 次 flop（10.9），仅需在上 $n\times n$ 子矩阵上运算的 Golub-Kahan 过程需要 $\frac{8}{3}n^3$ 次 flop. 总共运算计数为：

$$\text{LHC 双对角化的工作量：大约为 } 2mn^2+2n^3 \text{ 次 flop} \qquad (31.5)$$

对于 $m>\frac{5}{3}n$，这比 Golub-Kahan 双对角化更为经济（习题31.1）. 奇妙的是，LHC 方法先增加零然后再破坏它们（在 A 的上 $n\times n$ 方阵的下三角阵中），但最终是有好处的.

LHC 方法仅对 $m>\frac{5}{3}n$ 有益，但这一思想可以推广以便对任意的 $m>n$ 也能节省运算次数. 其技巧是不在计算的开始而在中间适当的地方应用 QR 因子分解. 由于在 Golub-Kahan 方法中，$m>n$ 的矩阵在实施双对角化时变得更小，所以这个技巧是有益的. 如果初始的纵横比为，比如，$m/n=3/2$，那它将稳定地增长到 5/3 和 2 以及更多. 在 k 步后，剩余矩阵的纵横比为 $(m-k)/(n-k)$，并且当这个数字变得充分大时，其意义为实施 QR 因子分解来把问题简化为方阵.

238

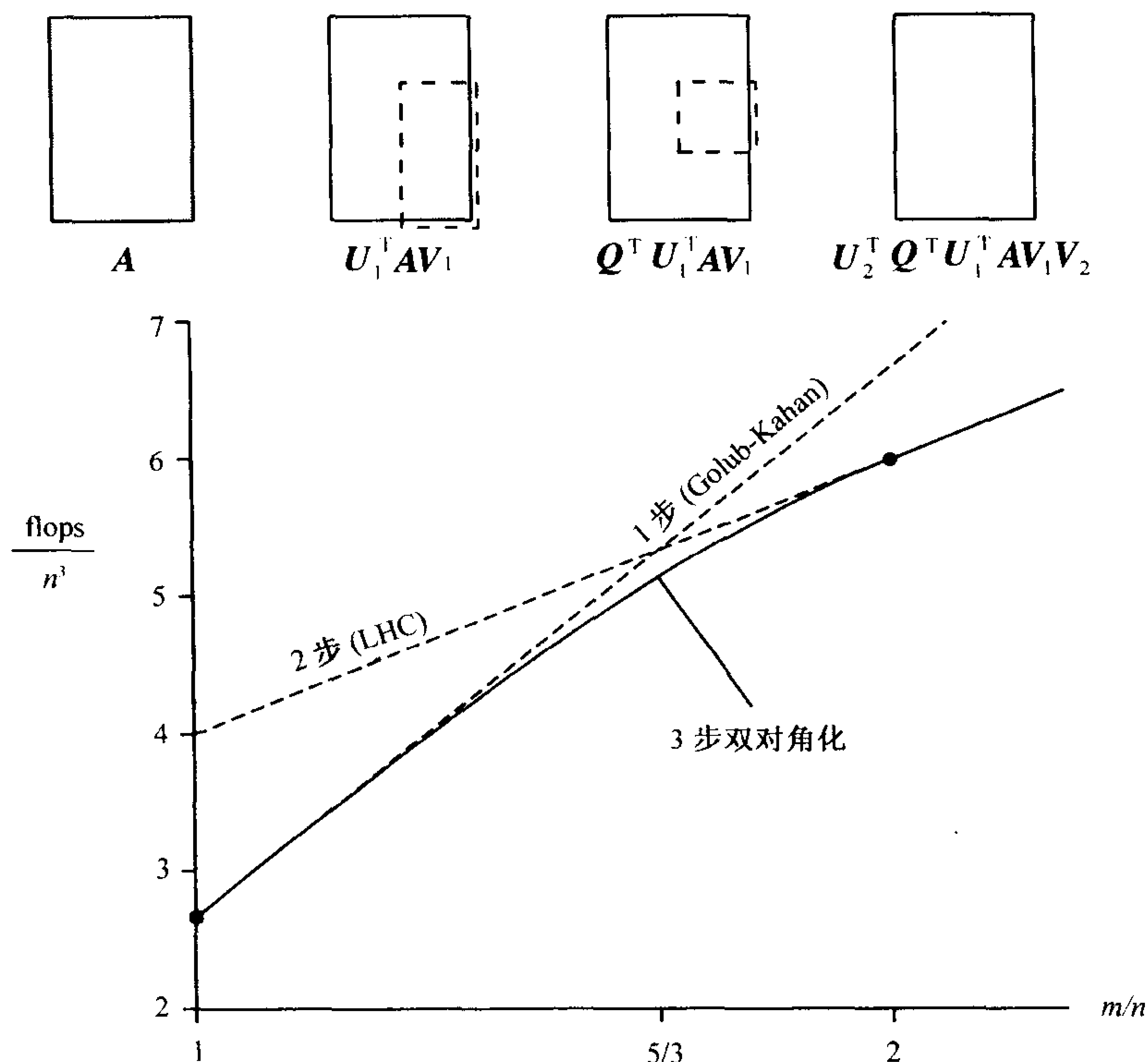


图 31-1 根据 (31.4)，(31.5) 和 (31.6) 的三个双对角化算法应用到 $m\times n$ 矩阵的运算次数. 3 步双对角化提供了另外两个方法之间良好的光消插值，尽管这个改进未必大

何时应该实施 QR 因子分解？如果目的仅为使计算量减到最小，那么这个回答是简单的——在纵横比达到 $(m-k)/(n-k)=2$ 时（习题31.2），这个选择导致了公式

步双对角化的工作量：大约为 $4mn^2-\frac{4}{3}n^3-\frac{2}{3}(m-n)^3$ 次 flop (31.6)

这是对于 $n<m<2n$ 的另外两个方法的适度改进.

三个方法的运算计数作为 m/n 的函数图示在图 31-1 中，必须承认，由 3 步方法完成的改进在实际中是很少的，除了运算计数外的其他因素可以确定在实际的机器用哪个方法上是最好的.

31.6 阶 段 2

在计算机 SVD 的阶段 2 中，要确定双对角矩阵 $\boldsymbol{B}$ 的 SVD. 从 20 世纪的 60 年代到 90 年代，对它的标准算法是 QR 算法的变形，近来，分而治之算法也具有竞争力了，今后它们很可能变成标准的方法. 在此不作详细讨论. 239

习 题

31.1 (a) 试证明，如在正文中的断言以及在图 31-1 中的说明，LHC 双对角化开始优先于 Golub-Kahan 双对角化的交叉纵横比是 $m/n=5/3$.

(b) 对于 $m/n=5/3$，3 步双对角化法比另外两个方法快出多大的比例？

31.2 证明，在 3 步双对角化中，实施 QR 因子分解的最优点是当矩阵达到纵横比为 2 时.

31.3 证明，如果双对角矩阵的两主对角线上元素均非零，那么矩阵的奇异值是相异的.（见习题 25.1）

31.4 设 $\boldsymbol{A}$ 为 $m\times m$ 上三角矩阵，其主对角线上元素为 0.1，对角线上每个元素为 1. 写出一个程序，用两种方法计算 $\boldsymbol{A}$ 的最小奇异值：调用标准 SVD 软件，并形成 $\boldsymbol{A}^*\boldsymbol{A}$，且计算其最小特征值的平方根. 对于 $1\leqslant m\leqslant 30$ 运行程序，在对数尺度下把结果图示为两条曲线. 此结果与这些算法的一般讨论一致吗？

31.5 设 $\boldsymbol{A}$ 为 $m\times n$ 矩阵，其元素是从 $N(0,1)$ 中独立抽取的样本，$N(0,1)$ 为均值为 0，方差为 1 的正态分布（与习题 12.3 作比较）. 设 $\boldsymbol{B}$ 为双对角矩阵

$$\boldsymbol{B}=\begin{bmatrix} x_m & y_{n-1} & & & \\ & x_{m-1} & y_{n-2} & & \\ & & \ddots & \ddots & \\ & & & x_{m-(n-2)} & y_1 \\ & & & & x_{m-(n-1)} \end{bmatrix},$$

其中每个 x 或 y 是从 χ^2 分布（自由度为附加下标）中独立抽取的样本的正平方根.［自由度为 k 的 χ^2 分布等于服从 $N(0,1)$ 的 k 个独立变量平方的和的分布.］

(a) 证明 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的奇异值分布是一样的.

(b) 用试验证明这个结果. 特别地，取 $m=100$ 和 $n=50$，用上面指出的办法构造随机矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$，并图示出相对于矩阵 $\boldsymbol{B}$ 的奇异值的矩阵 $\boldsymbol{A}$ 的奇异值. 240

第VI部分
迭　代　法

第32讲　迭代法综述

从本讲开始，本书的风格改变了，从非常熟悉的经典课题—直接法转到相对有活力的迭代法的领域. 有些方法很可能主导将来的大规模计算.

32.1　为什么要迭代?

线性代数中，迭代算法的重要性源于一个简单的事实：对于一般矩阵，非迭代的或“直接”算法所需的工作量达到 $O(m^3)$. 这太大了！在绝对意义下，当 m 很大时 m^3 非常巨大；在相对意义下，大多数矩阵问题输入仅包含 $O(m^2)$ 次数，而求解它们时所需的工作量为 $O(m^3)$，这看起来似乎不合理，由此在这两个意义下均太大了.

下表给出了这些年来矩阵计算的简史：

1950 : $m=20$	(Wilkinson)
1965 : $m=200$	(Forsythe 和 Moler)
1980 : $m=2\,000$	(LINPACK)
1995 : $m=20\,000$	(LAPACK)

在表中相应的年代，对于稠矩阵的直接矩阵计算，究竟什么维数可以认为是“大”，以上数据给出了一个大致的近似，例如，在20世纪60年代中期的“Forsythe 和 Moler 时代”（因其于1967年出版了一本有影响的教科书而得名），数百阶的矩阵就算大的了，这是因为在合理的时间内，在当时的机器上，数百阶的矩阵计算已达到了极限.

显然，在45年的过程中，可求解的矩阵问题的维数以 10^3 因子增加. 这个进展是巨大的，但在同一时期内，计算机硬件达到以 10^9 因子，即从浮点运算（每秒）到千兆浮点运算（每秒）的高速度发展，两者相比，前者显得失色得多. 从 10^9 是 10^3 的3次方这一事实看出，在历史上已经克服了直接矩阵算法的 $O(m^3)$ 难关.

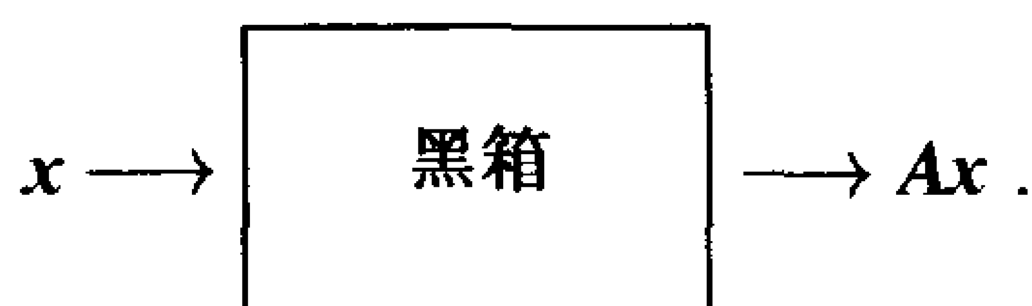
换句话说，如果矩阵问题能以 $O(m^2)$ 次运算来求解而不用 $O(m^3)$ 次运算，那么今天处理的某些矩阵的阶可以扩大10到100倍. 使用矩阵迭代方法对某些矩阵已经达到了这个目标.

32.2　结构、稀疏性和黑箱

当然，克服 $O(m^3)$ 难关不是件容易的事，事实上，对于“随机”矩阵问题几乎

是不可能的. 然而，实际中出现的大的矩阵问题远非随机矩阵问题，对此原因很简单. 小型矩阵，比如说3阶或30阶矩阵，可以直接由科学问题产生，其元素或多或少是任意的，这些科学问题如果表示结构中3个力之间的关系，或者表示化学反应中30种物质之间的关系. 而大型矩阵通常间接地由微分或积分方程离散化来得到. 可以说，如果 m 很大，它或许是 ∞ 的一个近似，因此，计算上感兴趣的大矩阵与它们可以指明的单个元素的巨大总数相比较要简单得多. 矩阵有某种类型的结构，而随着一年一年的过去，在愈来愈多的情况下已经找到了应用这种结构的方法.

大矩阵具有的最明显的结构是稀疏性，即零元素占优势，（与稀疏相对的是稠密.）例如，偏微分方程有限差分离散可以导出阶 $m=10^5$ 的矩阵，每行的非零元素的个数仅为 $\nu=10$. 这类结构已被将要讨论的迭代法利用，对于这些算法以黑箱（black box）形式使用矩阵：



迭代法仅需要有对任意 $\boldsymbol{x}$ 确定 $\boldsymbol{Ax}$ 的能力，其能力在计算机程序中由一个方法所影响，而此方法的内部作用必须与迭代算法的设计者无关.（一些迭代算法也需要计算 $\boldsymbol{A}^*\boldsymbol{x}$.）对于稀疏矩阵 $\boldsymbol{A}$ 的例子，容易设计一个方法来计算 $\boldsymbol{Ax}$ 仅需 $O(\nu m)$ 次运算而不是 $O(m^2)$ 次运算. 这与直接线性代数算法是有显著区别的，直接法，如高斯或豪斯霍尔德三角形化，为便于引入零而对矩阵元素进行了显式操作，这个过程一般破坏了矩阵的稀疏性.

244

历史上，稀疏性早已是迭代的矩阵计算中最常用的一种结构.（稀疏性也应用于快速直接方法，如嵌套剖分或极小度再排序，这些方法不在本书中讨论）更近地，另外类型的矩阵结构也可以在迭代矩阵计算中使用，这已变得很明显，即使包含的矩阵是稠密的也可使用. 例如，用数值方法解积分方程典型地导出了稠密的矩阵问题. 在工程中这些称为边界元或板块法. 这些矩阵的系数经常有大量的正则性在其中. 利用这些正则性的方法，如多极方法或小波展开是当今研究的活跃领域. 包含了实现这些方法的黑箱可以有几千条代码，其中所依据的思想较深，仅为专家所理解.

32.3　投影到克雷洛夫子空间

本书剩余部分所介绍的迭代法，是基于把 m 维问题投影到低维克雷洛夫子空间上这一思想. 给定矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$，相联系的克雷洛夫序列是向量 $\boldsymbol{b},\boldsymbol{Ab},\boldsymbol{A}^2\boldsymbol{b},\boldsymbol{A}^3\boldsymbol{b},\cdots$ 的集合，这些向量可以用黑箱以 $\boldsymbol{b},\boldsymbol{Ab},\boldsymbol{A}(\boldsymbol{Ab}),\boldsymbol{A}(\boldsymbol{A}(\boldsymbol{Ab})),\cdots$ 这样的形式计算出来. 对应的*克雷洛夫子空间*是由这些向量顺次组成的大集合所张成的.

具体地，将要讨论的算法排列在下表中：

	$Ax=b$	$Ax=\lambda x$
$A=A^*$	CG	兰乔斯
$A\neq A^*$	GMRES CGN BCG等	阿诺尔迪

(表中所填的是简称！例如 CG 表示共轭梯度，另外，要求 A 是正定的以及 A 是埃米尔特的.）在每一方法中，投影到克雷洛夫子空间的结果是把原来的矩阵问题约化到维数为 $n=1,2,3,\cdots$ 的矩阵问题的序列. 当 A 是埃米尔特矩阵时，约化的矩阵是三对角矩阵，而在非埃米尔特矩阵情形下，有海森伯格形式. 例如，用阿诺尔迪迭代逼近大矩阵的特征值，是用计算相继较大维数的某些海森伯格矩阵的特征值来完成的. [245]

32.4 步数、每步工作量和预处理

稠密线性代数的高斯消元法、QR 因子分解以及大多数其他的算法适合下面模型：有 $O(m)$ 步，每步需要 $O(m^2)$ 工作量，总的工作量估计为 $O(m^3)$.（当然，这些数字，特别是第 2 个数字，在并行计算机上可以改变.）对于迭代法，同样的数字仍可应用，但它们表示典型的最坏情况的状态. 这些方法要取得成功，往往是减少了其中一两种因素.

如果把术语"谱"作广义的解释，那么可以看到，对于收敛到满意精度所需要的步数典型地依赖于矩阵 A 的谱性质. 例如，如果 A 的特征值远离原点并聚集在一起，那么共轭梯度法保证了快速求解埃米尔特正定方程组 $Ax=b$. 类似地，如果实的埃米尔特矩阵的某些特征值与谱的其他元素是充分分离的，那么，兰乔斯迭代保证了快速计算这些特征值（并且如果开始迭代的初始向量是适当选取的）. 分析这些方法的收敛速度，依赖于数学领域的逼近论是具有吸引力的研究. 特别地，克雷洛夫子空间迭代算法的收敛性，与在实轴或复平面的子集上用多项式 $p(z)$ 逼近函数 $f(z)$ 的问题紧密相关.

在矩阵迭代中，每步的工作量主要依赖于矩阵的结构和在 $x\mapsto Ax$ 黑箱中这个结构的某些优点.

在线性代数中，理想的迭代法把步数从 $O(m)$ 缩减到 $O(1)$，每步工作量从 $O(m^2)$ 缩减到 $O(m)$，而总的工作量从 $O(m^3)$ 缩减到 $O(m)$. 在实际问题中确实发生过这样非常快的加速缩减，但大多数典型的改进多半是从 $O(m^3)$ 缩减到 $O(m^2)$. 在 20 世纪 90 年代中期，实际上大规模的工程计算中，迭代法是成功的，或许典型的结

果是它以 10 倍的量级胜过直接算法. 当机器发展更快，m 在将来变得更大，这个倍数还会增加并且迭代法变得更加重要，这说明了计算机科学的基本规律：计算机发展得越快，算法速度的重要性就越大.

32.5 真实解与近似解的比较

至少当实现了实际中感兴趣的迭代步数时，矩阵迭代方法原则上不会得到精确246的解而仅是近似解，甚至于没有舍入误差也是如此. 这个性质容易造成初学者对这些思想存有顾虑. 他们觉得迭代是没什么精致的“工程解”，而且怀疑其可靠性. 当知道这些方法较好时，顾虑就减少了. 最后，当在计算机上实现时，即使直接法也是不精确的：人们希望得到的回答是精确到机器的精度，而不需要更好. 由于迭代法可以达到机器精度的充分精确度，所以它们在理论上近似的需要就没有什么重要性了. 讲到精致，这里的思想是数值线性代数中几个最优秀的思想.

这些要点在图 32-1 中作了说明.

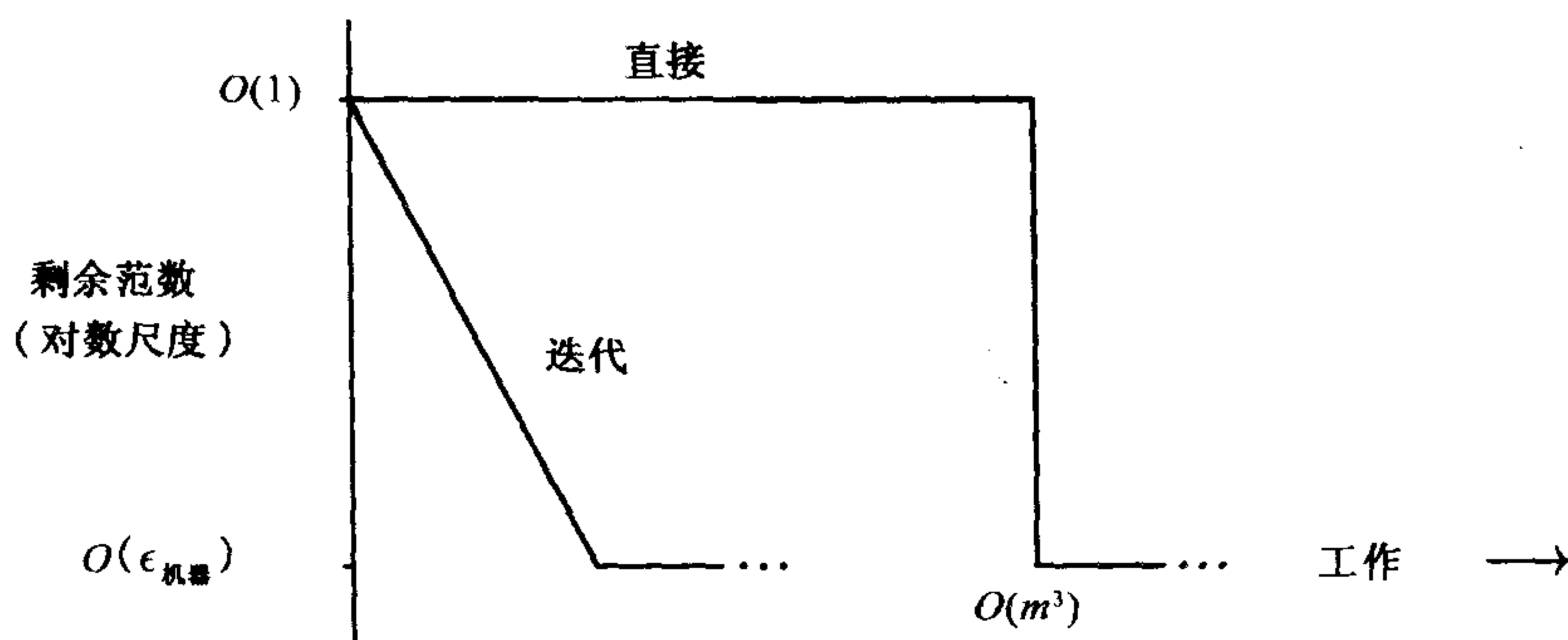


图 32-1　直接法和迭代法收敛性的图解说明. 在有利的情况下，迭代法在剩余达到$\epsilon_{机器}$阶之前是几何收敛. 直接法在 $O(m^3)$次运算完成之前没有一点进步，在该处剩余再次达到$\epsilon_{机器}$阶

32.6 克服 $O(m^3)$的直接法

最后，必须指出存在有限的，原则上是精确的直接算法，在求解稠密矩阵的 $Ax=b$和相关问题中，其运算量少于 $O(m^3)$. 这种类型的第一个算法是 Volker Strassen 于 1969 年发现的，他把高斯指数由 3 减少到 $\log_2(7)\approx 2.81$，随后的改进由 Coppersmith 和 Winograd 给出，他们把最有名的指数减少到现在的值，约为 2.376. 这些最著名指数的历史记录于图 32-2.

到目前为止，这些快速算法对于实际计算的影响由于下面的两个原因而被忽略

了. 其一是一般很少了解它们的稳定性质. 更基本的是，尽管在快速算法中的指数是惊人的，但是在开始胜过标准算法的 m 的交错值是非常大的. Strassen 的 $m^{2.81}$ 算法 247 可以对如 100 那样低的 m 胜过高斯消元法，但是因为 2.81 是如此接近 3，所以对于实际 m 值的成功决不会令人惊奇. 还有一些比这个指数更低的算法，但由于它包含着如此大的常数因子，以致对于当前计算机可以达到的 m 值的运算来说，这些算法比高斯消元法慢.

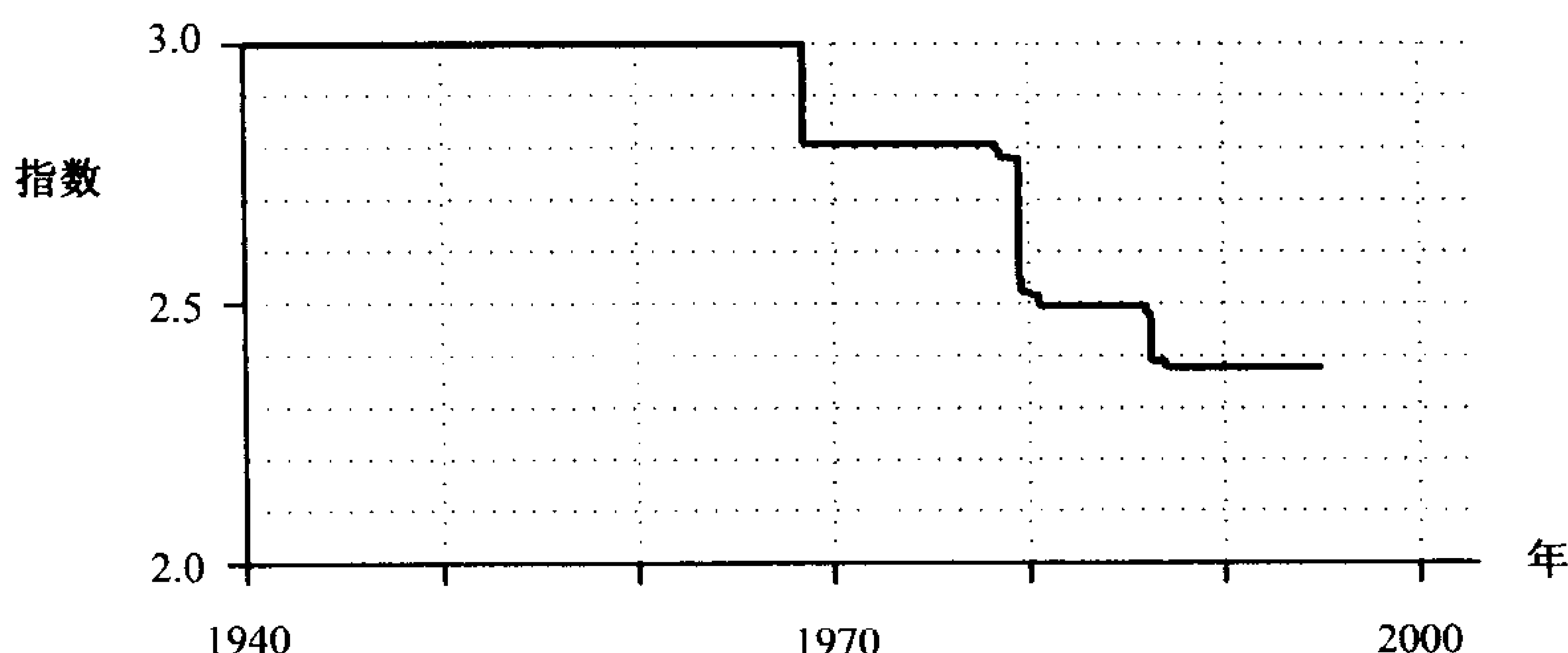


图 32-2 对于 $m\times m$ 矩阵的 $\boldsymbol{Ax}=\boldsymbol{b}$（或等价地计算 $\boldsymbol{A}^{-1}$，$\boldsymbol{AB}$ 或 $\det\boldsymbol{A}$），它的直接解最著名的指数作为时间的函数，直到 1968 年，最著名算法的复杂性为 $O(m^3)$. 目前最著名的算法以 $O(m^{2.376})$ 次 flop 来解 $\boldsymbol{Ax}=\boldsymbol{b}$，但常数相当大以致于这个算法不实用

但是在将来会发生什么呢? 实际情况是无人知道的. 很可能明天将有人发明"快速矩阵求逆"以 $m^2\log m$ 次 flop 来求解 $\boldsymbol{Ax}=\boldsymbol{b}$；或许你，读者在今天晚上就这样做了. 这样的发展将引发在数值计算历史上最大的变革.

习 题

32.1 用边界元方法离散三维椭圆型方程，其结果是大的稠密的线性方程组，其中每个方程对应于在大球上的三角曲面元素. 为改进精度，必须使三角形更小，因此增加了方程的个数. 但误差仅线性地减小，与最大三角形的直线 h 成比例.

选定 h 的值，用高斯消元法解方程组，并且在一分钟计算机时间内得到了 2 位数精度的解. 现需要解决 3 位数精度的解. 假设储存没有限制，在同样的计算机上实施新的计算，大约需要多少时间? 248

32.2 考虑块矩阵的乘积

$$\begin{bmatrix} \boldsymbol{W} & \boldsymbol{X} \\ \boldsymbol{Y} & \boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}\begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} \\ \boldsymbol{G} & \boldsymbol{H} \end{bmatrix},$$

其中，为简单起见，所有矩阵 $\boldsymbol{A},\boldsymbol{B},\cdots,\boldsymbol{Y},\boldsymbol{Z}$ 都假定为方阵并有同样的阶.

(a) 给定 $\boldsymbol{A},\boldsymbol{B},\cdots,\boldsymbol{G},\boldsymbol{H}$,用通常算法计算 $\boldsymbol{W},\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}$,要使用 (i) 多少矩阵加法? (ii) 多少矩阵乘法?

(b) Strassen 指出,$\boldsymbol{W},\boldsymbol{X},\boldsymbol{Y}$ 和 $\boldsymbol{Z}$ 也可以用下面的公式来计算.

$$\boldsymbol{P}_1=(\boldsymbol{A}+\boldsymbol{D})(\boldsymbol{E}+\boldsymbol{H}),\qquad \boldsymbol{P}_5=(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{H},$$
$$\boldsymbol{P}_2=(\boldsymbol{C}+\boldsymbol{D})\boldsymbol{E},\qquad \boldsymbol{P}_6=(\boldsymbol{C}-\boldsymbol{A})(\boldsymbol{E}+\boldsymbol{F}),$$
$$\boldsymbol{P}_3=\boldsymbol{A}(\boldsymbol{F}-\boldsymbol{H}),\qquad \boldsymbol{P}_7=(\boldsymbol{B}-\boldsymbol{D})(\boldsymbol{G}+\boldsymbol{H}),$$
$$\boldsymbol{P}_4=\boldsymbol{D}(\boldsymbol{G}-\boldsymbol{E}),$$
$$\boldsymbol{W}=\boldsymbol{P}_1+\boldsymbol{P}_4-\boldsymbol{P}_5+\boldsymbol{P}_7,\qquad \boldsymbol{Y}=\boldsymbol{P}_2+\boldsymbol{P}_4,$$
$$\boldsymbol{X}=\boldsymbol{P}_3+\boldsymbol{P}_5,\qquad \boldsymbol{Z}=\boldsymbol{P}_1+\boldsymbol{P}_3-\boldsymbol{P}_2+\boldsymbol{P}_6$$

现在包含了 (i) 多少矩阵加法或减法? (ii) 多少矩阵乘法?

(c) 试证明,循环地应用 Strassen 公式,可以得到阶为 $m=2^k$ 的矩阵乘法的算法,其运算次数当 $m\to\infty$ 时为 $O(m^{\log_2(7)})$.

249

(d) 写出一个实现这一思想的循环程序,并且运行程序来给出数值证明.

第 33 讲　阿诺尔迪迭代

尽管在克雷洛夫子空间矩阵迭代方面新的算法层出不穷，但这些算法有其共同的基础，就是几个基本的思想. 叙述这个基础可以用许多种方法. 我们将考虑阿诺尔迪过程，这是一种将一个矩阵变换到海森伯格型的格拉姆-施密特式的迭代。

33.1　一个比喻

假设，你被放逐到一个荒岛上，为消磨时间，你可以设法设计一个用正交相似变换把非埃米尔特矩阵约化到海森伯格型的算法，此正交相似变换从选定的第一列 q_1 开始逐列进行. 令人惊奇的是，或许你可以发现，用一个小时就可以解决此问题，并仍有时间来采集椰子当正餐. 你提供的方法称为阿诺尔迪迭代. 如果 $\boldsymbol{A}$ 是埃米尔特矩阵，那么海森伯格矩阵变为三对角矩阵，n 项递推关系式变为 3 项递推关系式. 阿诺尔迪迭代这一名称也改为兰乔斯迭代，这将在第 36 讲中讨论.

这是一个比喻. 为计算矩阵 $\boldsymbol{A}$ 的 QR 因子分解 $\boldsymbol{A}=\boldsymbol{QR}$，在本书中已讨论过两个方法：一个是用一系列正交运算使 $\boldsymbol{A}$ 三角形化的豪斯霍尔德镜射，另一个是用一系列三角形运算使 $\boldsymbol{A}$ 正交化的格拉姆-施密特正交化. 尽管在有舍入误差的情况下，豪斯霍尔德镜射产生了正交矩阵 $\boldsymbol{Q}$ 的一个很好的近似，但格拉姆-施密特过程有如下的好处：它可以在中途终止，得到 $\boldsymbol{A}$ 的前 n 列的约化 QR 分解. 计算矩阵 $\boldsymbol{A}$ 的豪斯霍尔德约化 $\boldsymbol{A}=\boldsymbol{QHQ}^*$ 的问题是完全类似的. 有两个标准的方法：豪斯霍尔德镜射（现在应用到 $\boldsymbol{A}$ 的两边而不是 $\boldsymbol{A}$ 的一边）和阿诺尔迪迭代. 于是，阿诺尔迪迭代是相似变换到海森伯格的格拉姆-施密特的模拟而不是 QR 因子分解，如格拉姆-施密特一样，阿诺尔迪迭代也有中途终止的优点，可以得到部分到豪斯霍尔德型的约化，以此可用各种方式来形成关于特征值或方程组的迭代算法.

250

这一讲与第 26 讲的关系正如第 8 讲与第 10 讲的关系一样.

可以把刚提到的 4 种算法总结在下表中：

	$\boldsymbol{A}=\boldsymbol{QR}$	$\boldsymbol{A}=\boldsymbol{QHQ}^*$
正交结构	豪斯霍尔德	豪斯霍尔德
结构的正交化	格拉姆-施密特	阿诺尔迪

在本书的剩余部分中，m 和 $n<m$ 是正整数，$\boldsymbol{A}$ 是实或复的 $m\times m$ 矩阵，$\|\cdot\|=\|\cdot\|_2$. 此外，一个更进一步的特征将戏剧性地出现，用 $\boldsymbol{b}$ 表示 m 维向量. 在阿诺尔迪过程中，用此向量作为初始向量，对于应用到特征值问题时，一般假定 $\boldsymbol{b}$ 是随机的. 对于应用到以后几讲中所讨论的方程组时，$\boldsymbol{b}$ 将是右端项，或更一般地，

$\boldsymbol{b}$ 为初始剩余向量（见习题35.5）.

33.2 阿诺尔迪迭代技巧

用正交相似变换把 A 完全约化到海森伯格型，可以写成 $A = QHQ^*$，或 $AQ = QH$. 然而，在处理迭代方法中，要考虑 m 非常大或是无限的情景，所以要计算完全约化是做不到的. 因此，考虑 $AQ = QH$ 的前面 n 列. 令 Q_n 是 $m \times n$ 矩阵，其列为 Q 的前 n 列：

251

$$Q_n = \left[\begin{array}{c|c|c|c} q_1 & q_2 & \cdots & q_n \end{array} \right]. \tag{33.1}$$

这里和在前面几讲中，它与本书其余部分的用法相一致，符号 Q_n 上面要加一帽子，但由于这些矩阵是长方形矩阵，为使公式不致混乱，所以没有加帽子.

令 $\tilde{H}_n$ 是 H 的 $(n+1) \times n$ 左上角部分，它也是海森伯格矩阵

$$\tilde{H}_n = \begin{bmatrix} h_{11} & & \cdots & h_{1n} \\ h_{21} & h_{22} & & \\ & \ddots & \ddots & \vdots \\ & & h_{n,n-1} & h_{nn} \\ & & & h_{n+1,n} \end{bmatrix}. \tag{33.2}$$

那么有

$$AQ_n = Q_{n+1} \tilde{H}_n \tag{33.3}$$

即，

$$\begin{bmatrix} \\ A \\ \\ \end{bmatrix} \left[\begin{array}{c|c|c} q_1 & \cdots & q_n \end{array} \right] = \left[\begin{array}{c|c|c} q_1 & \cdots & q_{n+1} \end{array} \right] \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ h_{21} & & \\ & \ddots & \vdots \\ & & h_{n+1,n} \end{bmatrix}.$$

这个方程组的第 n 列可以写为如下：

$$Aq_n = h_{1n} q_1 + \cdots + h_{nn} q_n + h_{n+1,n} q_{n+1}. \tag{33.4}$$

总之，q_{n+1}满足包含其本身和前面克雷洛夫向量的 $(n+1)$ 项递推关系.

阿诺尔迪迭代仅是实现（33.4）的修正格拉姆-施密特迭代. 下面的算法应与算法 8.1 作比较.

算法 33.1 阿诺尔迪迭代

b = 任意值, $q_1 = b/\|b\|$
for $n = 1, 2, 3\cdots$
  $v = Aq_n$
  for $j = 1$ **to** n
    $h_{jn} = q_j^* v$
    $v = v - h_{jn} q_j$
  $h_{n+1,n} = \|v\|$ [见练习 33.2 $h_{n+1,n} = 0$]
  $q_{n+1} = v/h_{n+1,n}$

252

容易看出，阿诺尔迪过程是很简单的．在如 MATLAB 这样的高级语言中，阿诺尔迪过程用不到 12 行就可以完成了．矩阵 $\boldsymbol{A}$ 仅出现在乘积 $\boldsymbol{A}\boldsymbol{q}_n$ 中，而 $\boldsymbol{A}\boldsymbol{q}_n$ 可以用上一讲描述的黑箱过程来计算．

33.3 克雷洛夫矩阵的 QR 因子分解

阿诺尔迪过程的强大在于对其可以作出各种解释以及这些解释所蕴含的算法．下面给出我们的第 1 个解释，考虑递推式（33.4），显然，从这个公式可以得出，向量$\{\boldsymbol{q}_j\}$形成了由 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 产生的逐次*克雷洛夫子空间*（Krylov subspace）的基，它们定义如下：

$$\mathcal{K}_n = \langle \boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \cdots, \boldsymbol{A}^{n-1}\boldsymbol{b} \rangle = \langle \boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_n \rangle \subseteq \mathbb{C}^m \tag{33.5}$$

此外，由于向量 $\boldsymbol{q}_j$ 是正交的，所以这是正交基．于是，阿诺尔迪过程描述为逐次克雷洛夫子空间的正交基的系统构造．

为了用矩阵形式来表示这一见解，定义 $\boldsymbol{K}_n$ 为 $m \times n$ *克雷洛夫矩阵*（Krylov matrix）

$$\boldsymbol{K}_n = \left[\begin{array}{c|c|c|c} \boldsymbol{b} & \boldsymbol{A}\boldsymbol{b} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{b} \end{array}\right]. \tag{33.6}$$

那么，K_n 必有约化的 QR 因子分解

$$\boldsymbol{K}_n = \boldsymbol{Q}_n \boldsymbol{R}_n, \tag{33.7}$$

其中 $\boldsymbol{Q}_n$ 是如同上面一样的矩阵．在阿诺尔迪过程中，$\boldsymbol{K}_n$ 和 $\boldsymbol{R}_n$ 都不能显式地形成．这样做将形成不稳定的算法，其原因是当 $\boldsymbol{K}_n$ 的列趋于逼近 $\boldsymbol{A}$ 的同一主特征向量时，一般都存在高度病态的矩阵．然而，（33.6）和（33.7）给出了为什么阿诺尔迪过程得到了决定某些特征值的有效方法的一个直观解释．显然，可以预期 $\boldsymbol{K}_n$ 包含了关于

253

$\boldsymbol{A}$ 的最大模特征值的充分多的信息，并且也可以预期 QR 因子分解可以用以优势特征向量开始，相继分离出近似特征向量的方法显示这些信息.

刚才给出的解释可以联想到本书中更早出现的类似讨论.（33.6）~（33.7）和阿诺尔迪算法之间的关系类似于计算矩阵特征值的同时迭代和 QR 算法之间的关系. 一个容易理解但不稳定，而另一个较难捉摸，但较为稳定. 其差别是，阿诺尔迪迭代基于其列为 $\boldsymbol{b},\boldsymbol{Ab},\cdots,\boldsymbol{A}^{n-1}\boldsymbol{b}$ 的矩阵的 QR 因子分解（33.7），而同时迭代和 QR 算法基于其列为 $\boldsymbol{A}^n\boldsymbol{e}_1,\cdots,\boldsymbol{A}^n\boldsymbol{e}_m$ 的矩阵的 QR 因子分解（28.16）. 这些类似总结成下表：

	拟-直接	迭代
直接但不稳定	同时迭代	(33.6) ~ (33.7)
难理解但稳定	QR 算法	阿诺尔迪

33.4 在克雷洛夫子空间上的投影

也可将阿诺尔迪过程看作在逐次克雷洛夫子空间上投影的计算. 为此，注意到乘积 $\boldsymbol{Q}_n^*\boldsymbol{Q}_{n+1}$ 是 $n\times(n+1)$ 单位阵，即 $n\times(n+1)$ 矩阵，其主对角线上元素为 1，其余为 0. 因此 $\boldsymbol{Q}_n^*\boldsymbol{Q}_{n+1}\tilde{\boldsymbol{H}}_n$ 是消去 $\tilde{\boldsymbol{H}}_n$ 的最后一行得到的 $n\times n$ 海森伯格矩阵：

$$\boldsymbol{H}_n=\begin{bmatrix} h_{11} & & \cdots & h_{1n} \\ h_{21} & h_{22} & & \\ & \ddots & \ddots & \vdots \\ & & h_{n,n-1} & h_{nn} \end{bmatrix}. \tag{33.8}$$

因此从（33.3）有

$$\boldsymbol{H}_n=\boldsymbol{Q}_n^*\boldsymbol{A}\boldsymbol{Q}_n. \tag{33.9}$$

这个矩阵可以解释为 $\boldsymbol{A}$ 在 $\mathcal{K}_n$ 上的正交投影在基 $\{\boldsymbol{q}_1,\cdots,\boldsymbol{q}_n\}$ 上的表示. 这个解释的意思清楚吗？下面作精确的陈述. 考虑定义如下的 $\mathcal{K}_n\to\mathcal{K}_n$ 线性算子：给定 $v\in\mathcal{K}_n$，把 $\boldsymbol{A}$ 作用到它上，那么正交投影 $\boldsymbol{A}\boldsymbol{v}$ 反到空间 $\mathcal{K}_n$. 由于 $\mathbb{C}^m$ 在 $\mathcal{K}_n$ 上的正交投影是 $\boldsymbol{Q}_n\boldsymbol{Q}_n^*$，对于 $\mathbb{C}^m$ 的标准基，这个算子可以写成 $\boldsymbol{Q}_n\boldsymbol{Q}_n^*\boldsymbol{A}$. 因此，对于 $\boldsymbol{Q}_n$ 的列的基，它可以写成 $\boldsymbol{Q}_n^*\boldsymbol{A}\boldsymbol{Q}_n$.

刚才讨论的这类投影流行于应用领域和数值数学中. 在别处也称为瑞利-里茨方法. 由于 $\boldsymbol{H}_n$ 的对角元素可以看成 $\boldsymbol{A}$ 关于向量 $\boldsymbol{q}_j$ 的瑞利商，所以并不一致. 这个投影方法也是构成解偏微分方程的有限元方法基础的思想，同样也是更近的称为谱方法的思想.

254

由于 $\boldsymbol{H}_n$ 是 $\boldsymbol{A}$ 的投影，所以可以设想 $\boldsymbol{H}_n$ 的特征值将以有效形式与 $\boldsymbol{A}$ 的一些特征

值有关. 这 n 个数.

$$\{\theta_j\} = \{H_n \text{ 的特征值}\}, \tag{33.10}$$

称为 A 的阿诺尔迪特征值估计（在 n 步）或 A 的里茨值（关于 $\mathcal{K}_n$）. 在下一讲中，将看到某些这样的数可以是 A 的某些特征值的极其精确的近似，甚至对于 $n \ll m$ 也一样.

这讲的结论总结成一个定理，可与定理 28.3 作比较.

定理 33.1 阿诺尔迪迭代产生的矩阵 Q_n 是克雷洛夫矩阵（33.6）的约化 QR 因子：

$$K_n = Q_n R_n. \tag{33.11}$$

豪斯霍尔德矩阵 H_n 是相应的投影

$$H_n = Q_n^* A Q_n. \tag{33.12}$$

并且逐次迭代与下面的公式有关

$$AQ_n = Q_{n+1} \tilde{H}_n. \tag{33.13}$$

习 题

33.1 设 $A \in \mathbb{C}^{m \times m}$，$b \in \mathbb{C}^m$ 是任意给定的. 证明，对任意的 $x \in K_n$ 等于 $p(A)b$，其中 p 为次数小于等于 $n-1$ 的某一多项式.

33.2 假设对于特定的 A 和 b，运行算法 33.1 直到 n 步，在此步元素 $h_{n+1,n}=0$.

(a) 指出在此情况下怎样来化简（33.13）. 此情况可以推出 A 的完全 $m \times m$ 豪斯霍尔德约化 $A=QHQ^*$ 的结构是什么?

(b) 证明 $\mathcal{K}_n$ 是 A 的不变子空间，即 $A\mathcal{K}_n \subseteq \mathcal{K}_n$.

(c) 证明用 b 产生的 A 的克雷洛夫子空间若是如（33.5）用 $\mathcal{K}_n = \langle b, Ab, \cdots, A^{k-1}b \rangle$ 来定义的，那么 $\mathcal{K}_n = \mathcal{K}_{n+1} = \mathcal{K}_{n+2} = \cdots$.

(d) 证明 H_n 的每一特征值是 A 的一个特征值.

(e) 证明如果 A 是非奇异的，那么方程组 $Ax=b$ 的解 x 在 $\mathcal{K}_n$ 中.

255

元素 $h_{n+1,n}=0$ 的出现称为阿诺尔迪迭代中断（breakdown），但这是一个良性中断. 在计算特征值（第 34 讲）或解方程组（第 35 讲）的应用中，由于（d）和（e），中断通常意味着收敛出现和迭代终止. 另一方面，新的正交向量 q_{n+1} 可以随机地选取，那么迭代可以继续.

33.3 (a) 假定算法 33.1 对于特殊的 A 和 b 执行了，并且运行到完成（$n=m$），没有在上一习题中所描述的中断的类型. 证明此情况推出 A 的极小多项式是 m 次的.

(b) 相反地，假设 A 的极小多项式是 m 次的，证明，对于特殊选取的 b，算法 33.1 不一定能运行到完成.

(c) 说明为什么（a）的结果与习题 25.1（b）不矛盾.

256

第34讲　用阿诺尔迪迭代求特征值

阿诺尔迪迭代可以看作：许多数值线性代数迭代算法的基础；更为特殊地，求非埃米尔特矩阵特征值的技巧. 这里考虑第二个，即其特殊作用，描术它与非常重要的多项式逼近理论的联系.

34.1　用阿诺尔迪迭代计算特征值

用阿诺尔迪迭代计算特征值如下进行. 迭代实施如上一讲所描述的（算法33.1），在每步 n，或在不多的步上，海森伯格矩阵 H_n 的特征值是用标准方法，例如QR算法，来计算的.（在实际中，这意味着调用如由EISPACK，LAPACK或MATLAB提供的软件.）这些是“阿诺尔迪估计”或“里茨值”（33.10）. 从其中的一些估计数值中可以看出它们是快速收敛的，且经常是几何收敛（即线性收敛，见习题25.2），当上述情况发生时，那么有相当的把握设想收敛的值是 A 的特征值.

由于对于可行的计算有 $n \ll m$，所以，当然不能要求用这个方法计算 A 的所有特征值. 那么，阿诺尔迪迭代求出的是哪个特征值呢？一般地，阿诺尔迪迭代求出极（extreme）特征值，即接近于 A 的谱的边缘的特征值. 幸而，这些正是在大多数应用中主要感兴趣的特征值.

257

例如，在流体动力学的稳定性问题中，目的是要确定，对于光滑流体的小扰动是否会不稳定地增长，在此情况下，流动很可能分解为另一种更为复杂的形式，或许是湍流. 可用下面的一般方法来确定稳定性，先把问题线性化，得到控制小扰动发展的雅可比算子，然后在复平面中计算这个算子的最右特征值，即具有最大实部的特征值. 如果这个特征值在右半平面内，那么流动不稳定，而若这个特征值在左半平面内，那么流动是稳定的.

已经设计出了关于阿诺尔迪方法的许多变形，例如，对于在复平面的某一部分比其他部分有更多重要性的问题的加速设计. 在此只限于关注原始的阿诺尔迪方法，并试图给出一个思想，说明为什么这种方法会收敛，为什么这种方法有助于找出 A 的极端特征值，以及如何更快地求出它们.

34.2　警告的注解：非正规性

首先要读者注意一个警告. 非埃米尔特矩阵的特征值的物理意义有时没有想像的那样大. 如果矩阵极不正规（即如果其特征向量极不正交，它隐含着其特征值是病态的），那么特征值对由矩阵实际性态确定的物理系统影响很小（习题

24.3 和 26.2). 例如，在流体流过一个圆管的问题中，线性化方程的雅可比算子在左半平面内有其全部特征值，这暗示了该流动应是稳定的. 然而，实际情形中，流体高速流过圆管总是湍流. 这个自相矛盾现象的解释基于如下事实：虽然线性化问题有着完全衰减的特征模式，但它仍然把某些非模态流扰动放大了许多个数量级. 这样的现象不可能发生在正规矩阵或算子中，例如对称的，埃米尔特或自伴的矩阵或算子.

如果答案对于扰动是非常敏感的，那么很可能问了错误的问题. 要强调，任何人在面临计算包含高灵敏度特征值的非埃米尔特特征值时应记住这个原则. 如果你是一个数值分析工作者，问题是由在科学或工程中的同事提供的，那么你不能承接这样的问题，除非你确认你的同事保证这确实是真实物理问题的特征值，甚至关于矩阵元素扰动的条件数是 10^4. 也许存在着非正规算子的高度灵敏的特征值有真正的物理意义的这种情况，但是更多的情况是，当特征值需要真正深刻分析时，恰恰进行了错误的研究.

258

34.3 阿诺尔迪和多项式逼近

令 $\boldsymbol{x}$ 是克雷洛夫子空间 $\mathcal{K}_n$ (33.5) 中的向量，这样一个向量可以写成 $\boldsymbol{A}$ 的幂乘以 $\boldsymbol{b}$ 的线性组合：

$$\boldsymbol{x} = c_0\boldsymbol{b} + c_1\boldsymbol{A}\boldsymbol{b} + c_2\boldsymbol{A}^2\boldsymbol{b} + \cdots + c_{n-1}\boldsymbol{A}^{n-1}\boldsymbol{b}. \tag{34.1}$$

这个表达式有一个紧凑描述：它是 $\boldsymbol{A}$ 的*多项式*乘以 $\boldsymbol{b}$，即如果 q 是多项式 $q(z) = c_0 + c_1 z + \cdots + c_{n-1}z^{n-1}$，那么有

$$\boldsymbol{x} = q(\boldsymbol{A})\boldsymbol{b}, \tag{34.2}$$

这已在习题 33.1 中指出过. 由于形如 (34.1) 的每个向量可以表示为 (34.2) 形式，所以克雷洛夫子空间迭代经常可以用矩阵多项式来分析.

阿诺尔迪迭代的一个分析可以取成下面的形式，定义

$$P^n = \{n \text{ 次首 1 多项式}\}.$$

（“首 1”意思是 n 次项的系数是 1.）现在考虑下面的逼近问题.

> **阿诺尔迪/兰乔斯逼近问题.** 求 $p^n \in P^n$ 使得
> $$\|p^n(\boldsymbol{A})\boldsymbol{b}\| = \text{极小值}. \tag{34.3}$$

本书的这一部分总是 $\|\cdot\| = \|\cdot\|_2$. 对于 P^n 用上标的原因是，在下一讲中，将考虑用 $c_0 = 1$ 而不是用 $c_n = 1$ 正规化 n 次多项式空间，对这个空间将用下标.

阿诺尔迪迭代有显著的性质，即可精确地解 (34.3).

定理34.1　只要阿诺尔迪迭代不中断（即$\boldsymbol{K}_n$是满秩n），那么（34.3）有惟一解p^n，即$\boldsymbol{H}_n$中的特征多项式.

证明　首先，注意到如果$p\in P^n$，那么向量$p(\boldsymbol{A})\boldsymbol{b}$可以写成，对于某一$\boldsymbol{y}\in\mathbb{C}^n$，$p(\boldsymbol{A})\boldsymbol{b}=\boldsymbol{A}^n\boldsymbol{b}-\boldsymbol{Q}_n\boldsymbol{y}$，其中$\boldsymbol{Q}_n$由（33.1）定义. 换言之，（34.3）等价于一个线性最小二乘问题：在$\mathcal{K}_n$中求一个最靠近$\boldsymbol{A}^n\boldsymbol{b}$的点，或用矩阵术语，求$y$使得$\|\boldsymbol{A}^n\boldsymbol{b}-\boldsymbol{Q}_n\boldsymbol{y}\|$是最小的. 此解用正交性条件$p^n(\boldsymbol{A})\boldsymbol{b}\perp\mathcal{K}_n$来表征（其说明见图34-1），或等价地，$\boldsymbol{Q}_n^*p^n(\boldsymbol{A})\boldsymbol{b}=0$. 现考虑在上一讲开始提到的因子分解$\boldsymbol{A}=\boldsymbol{QHQ}^*$. 在阿诺尔迪过程的第$n$步，已经计算了$\boldsymbol{Q}$的前$n$列和$\boldsymbol{H}$；于是可以知道，存在一个这种类型的因子分解

259

$$\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{Q}_n & \boldsymbol{U}\end{bmatrix},\quad \boldsymbol{H}=\begin{bmatrix}\boldsymbol{H}_n & \boldsymbol{X}_1\\ \boldsymbol{X}_2 & \boldsymbol{X}_3\end{bmatrix}$$

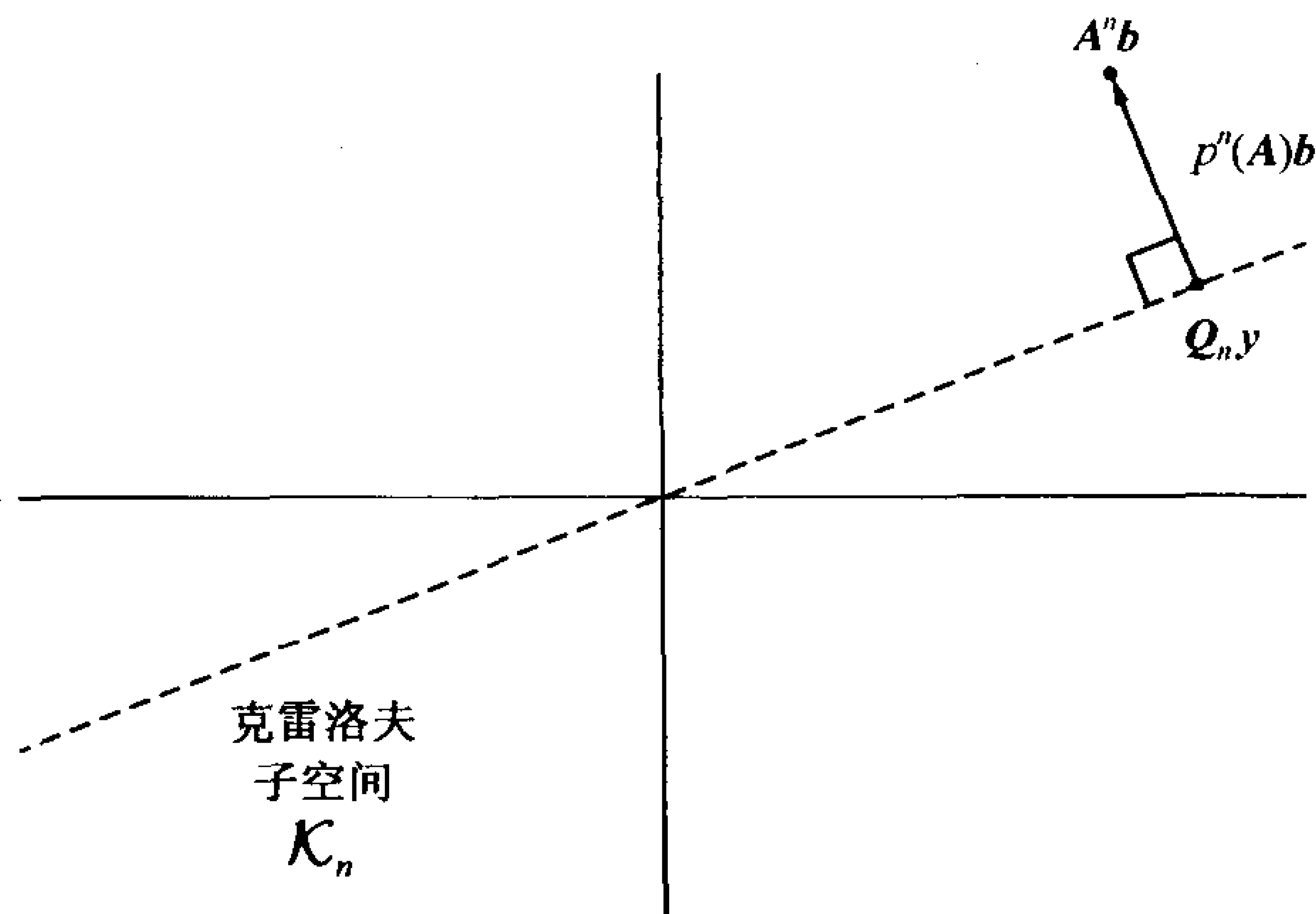


图34-1　基于阿诺尔迪迭代的最小二乘多项式逼近问题

其中$\boldsymbol{U}$为$m\times(m-n)$矩阵，其列标准正交，并与$\boldsymbol{Q}_n$的列也正交，$\boldsymbol{X}_1,\boldsymbol{X}_2$和$\boldsymbol{X}_3$的维数分别为$n\times(m-n)$，$(m-n)\times n$和$(m-n)\times(m-n)$，其元素除$\boldsymbol{X}_2$的右上部分外均为零. 正交性条件变成$\boldsymbol{Q}_n^*\boldsymbol{Q}p^n(\boldsymbol{H})\boldsymbol{Q}^*\boldsymbol{b}=\boldsymbol{0}$，这也相当于条件：$p^n(\boldsymbol{H})$的第1列的前$n$个元素是零（因仅$\boldsymbol{Q}^*\boldsymbol{b}$的第一个元素为非零，见第35讲）. 因为$\boldsymbol{H}$的结构，这些也是$p^n(\boldsymbol{H}_n)$第1列的前$n$个元素. 用Cayley-Hamilton定理，如果p^n是$\boldsymbol{H}_n$的特征多项式，那么它们是零. 相反地，假设有另外的多项式p^n满足$p^n(\boldsymbol{A})\boldsymbol{b}\perp\mathcal{K}_n$，取它们的差，将给出$n-1$次非零多项式$\boldsymbol{q}$满足$\boldsymbol{q}(\boldsymbol{A})\boldsymbol{b}=0$，这就破坏了$\boldsymbol{K}_n$是满秩这一假设.　□

定理34.1给出了用阿诺尔迪迭代产生里茨值的一个新解释，它们是最优多项式（34.3）的基础.

34.4　不变性性质

定理 34.1 提供了一个容易的方法来记住阿诺尔迪迭代的某些基本性质. 例如，由于首 1 多项式族 P^n 关于平移 $z \mapsto z+\alpha$ 是不变的，所以阿诺尔迪迭代也是平移不变的. 不同于以后将要讨论的求解 $Ax=b$ 的迭代算法，例如 GMRES，阿诺尔迪迭代并不"知道原点在何处".

260

下面是这个和其他不变性性质的总结. 这个定理的每一部分是定理 34.1 的推论，将不给出其证明.

定理 34.2　*设阿诺尔迪迭代应用到矩阵 $A \in \mathbb{C}^{m\times m}$，方法如上面所描述.*

平移不变性. *如果对于某一 $\sigma \in \mathbb{C}$，A 变换到 $A+\sigma I$，b 保持不变，那么在每步里茨值 $\{\theta_j\}$ 变换到 $\{\theta_j+\sigma\}$.*

标度不变性. *如果对于某一 $\sigma \in \mathbb{C}$，A 变换到 σA，并且 b 保持不变，那么里茨值 $\{\theta_j\}$ 变换到 $\{\sigma\theta_j\}$.*

酉相似变换下的不变性. *如果对于某一酉矩阵 U，A 变换到 UAU^*，并且 b 变换到 Ub，那么里茨值 $\{\theta_j\}$ 保持不变.*

在全部 3 种情形中，里茨向量，即为对应于特征向量 $y_j \in H_n$ 的向量 $Q_n y_j$，在所指出的变换下是不变的.

利用定理 24.9，每个矩阵 A 酉相似于上三角形矩阵. 于是在酉相似变换下的不变性的性质隐含着，为了理解阿诺尔迪迭代的收敛性质，原则上仅考虑上三角形矩阵就足够了. 然而，仅考虑对角矩阵是不够的. 在非埃米尔特（非正规）矩阵情况下，矩阵比其特征值有更多的信息.

34.5　阿诺尔迪如何求特征值

现在可以开始谈及在这一讲的标题中提出的问题了. 定理 34.1 断言，实际上，阿诺尔迪迭代的"目的"是解多项式逼近问题，或等价地解包含克雷洛夫子空间的最小二乘问题. 如果阿诺尔迪迭代对求特征值有帮助，那么它必须是达到这一目的的副产品.

多项式逼近必须用 A 的特征值来做什么？稍作考虑就可指出，按下面的方法这两者之间有一个联系. 如果要求一个多项式 p^n，它满足 $p^n(A)$ 很小这一性质，那么有效的办法是选择多项式 p^n 使其零点靠近 A 的特征值.

考虑一个极端情形. 假设 A 是可对角化的并仅有 $n \ll m$ 个不同的特征值，因此有 n 次极小多项式. 从定理 34.1 可以清楚得出，在 n 步之后，至少如果向量 b 包含了与每个特征值有关的方向上的分量，那么全部这些特征值将被精确地求出. 于是，在 n 步之后，阿诺尔迪迭代已经精确地计算了 A 的极小多项式.

在实际应用中，有大量同样的现象发生，然而，现在里茨值与特征值的一致性

是近似的代替了正确的，并且代替了极小多项式，其结果是“伪极小多项式”，即多

261 项式 p^n 使得 $\|p^n(A)\|$ 是小的.

34.6 阿诺尔迪双纽线

这个收敛过程可以用在复平面内画出双纽线来作图示说明. **双纽线**（lemniscate）是一曲线或曲线的集合.

$$\{z\in\mathbb{C}:|p(z)|=C\} \tag{34.4}$$

其中 p 是一个多项式，C 为实常数. 如果 p 是关于矩阵 A 的阿诺尔迪迭代在 n 步的阿诺尔迪多项式 p^n，C 取值为

$$C=\frac{\|p^n(A)b\|}{\|b\|}, \tag{34.5}$$

那么由（34.4）定义的曲线可以称为**阿诺尔迪双纽线**. 当迭代数 n 增加时，这些双纽线的分支典型地以围绕 A 的极特征值出现，然后很快收缩到一个点，即是特征值本身.

为说明这些思想，设 A 为维数 $m=100$ 的一个方阵，其元素是来自于均值为 0 和标准差为 $m^{-\frac{1}{2}}$ 的实正态分布的独立随机数（习题 12.3）. 由于 A 是实的，所以其特征值是由实数和复共轭对所组成. 标准差的选择是使得特征值在单位圆盘 $|z|\leqslant 1$ 内近似地均匀分布. 但现在把角元素 a_{11} 变为 1.5 来建立一个分离特征值，见图 34-2.

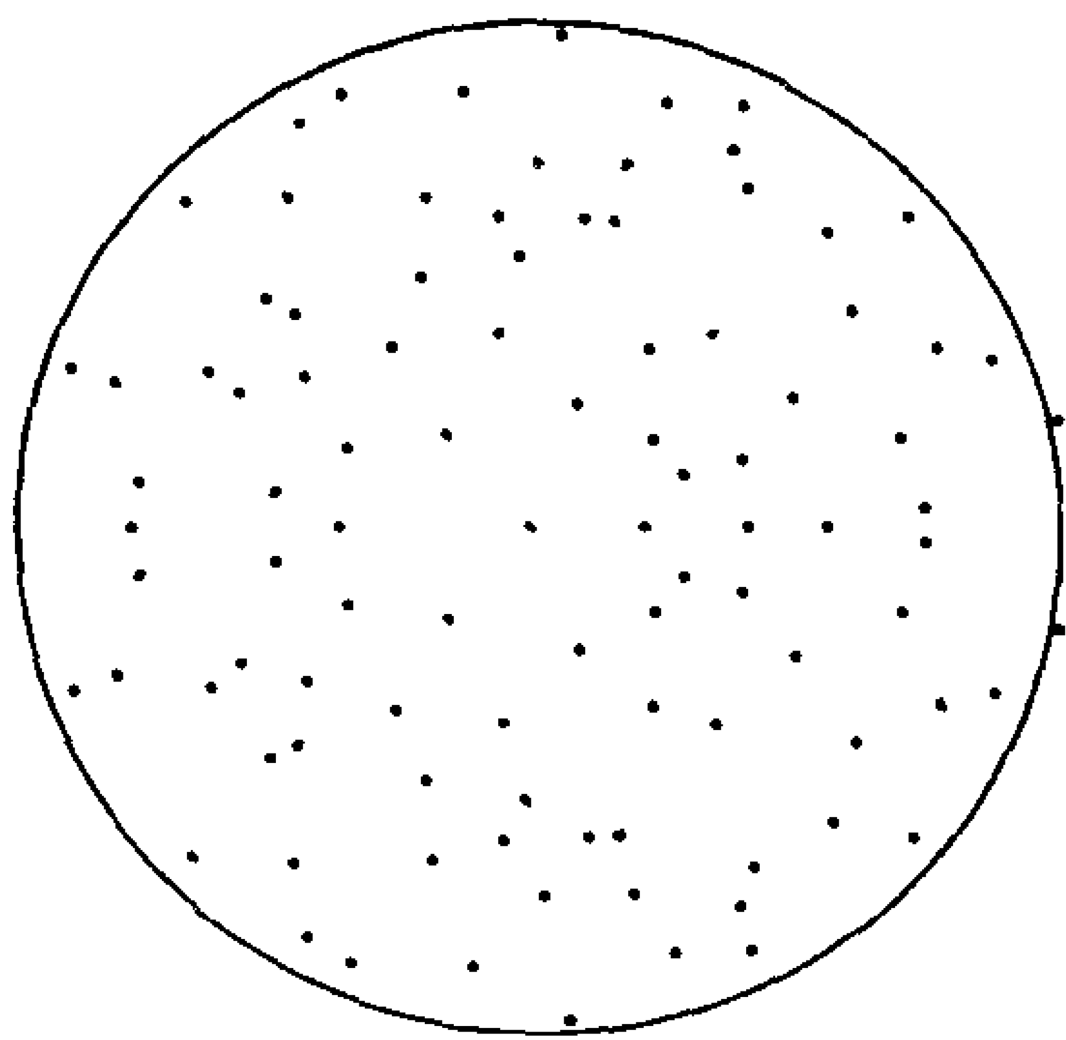


图 34-2　100×100 矩阵 A 的特征值，矩阵 A 的元素除（1,1）位置外均是随机的. 圆是 $\mathbb{C}$ 中的单位圆，除分离特征值 $\lambda\approx1.4852$ 外，所有特征值在单位圆盘内近似地均匀分布

把阿诺尔迪迭代应用到这一矩阵，初始向量取为随机向量 q_1. 在 n 步，矩阵 H_n 已经形成，其特征多项式反映了迭代的关于 A 的谱的当前知识. 图 34-3 画出了在步 $n=5,6,7,8$ 的阿诺尔迪双纽线. 在 $n=1$（没有指出），双纽线是一个精确的圆. 但稍微受到分离特征值的影响. 在 $n=5$ 之前，圆开始在 λ 方向凸出. 在 $n=6$ 时，凸出部分箍断并且出现了双纽线的一个新分支. 然后，这个分支继续下去，在其后的每

次迭代中收缩.

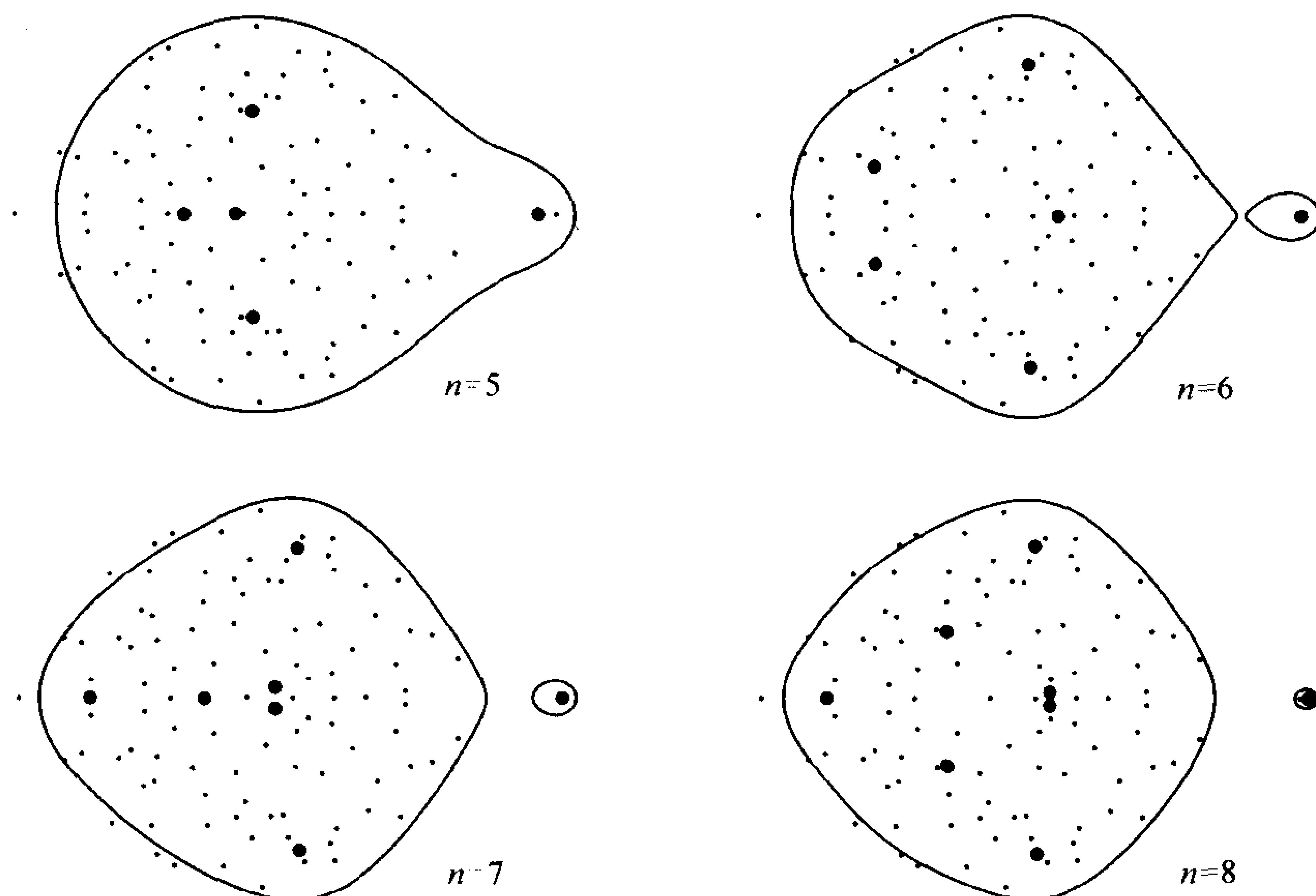


图 34-3　对于同样的矩阵 $\boldsymbol{A}$，在步 $n=5,6,7,8$ 上阿诺尔迪双纽线（34.4）~（34.5）. 小的点是 $\boldsymbol{A}$ 的特征值；大的点是 $\boldsymbol{H}_n$ 的特征值，即里茨值. 阿诺尔迪双纽线的一个分支首先“吞下”分离特征值，在其后的迭代中，它以几何速率收缩到一个点

34.7　几何收敛

在某些情况下，对于 $\boldsymbol{A}$ 的特征值，一些阿诺尔迪特征值估计的收敛是几何的. 这些问题在当前是不能完全被理解的，并在此不引用理论结果. 相反，仅用上面的数值例子考察其收敛，并给出部分说明.

从如图 34-3 的一个图上，不能确定收敛速率超过精度的 1 或 2 位数字. 对于如前同样的例子，用画出 $|\lambda^{(n)}-\lambda|$ 作为 n 的函数曲线的图 34-4 来填补这个差距，其中 $\lambda^{(n)}$ 是在 n 步上最靠近离群特征值 λ 的阿诺尔迪特征值估计. 注意到的第一件事情是收敛显然是几何的. 在 50 次迭代之后，达到了精度的 12 位数字. 如果 $\boldsymbol{A}$ 的维数用 1000 来代替 100，那么这个图形也没有太多的不同. 262 ~ 263

可以作更多的定量考察，不论怎样，对于最初几十次迭代，至少在图 34-3 中的收敛近似于速率.

$$|\lambda^{(n)}-\lambda|\approx\left(\frac{2}{3}\right)^n. \tag{34.6}$$

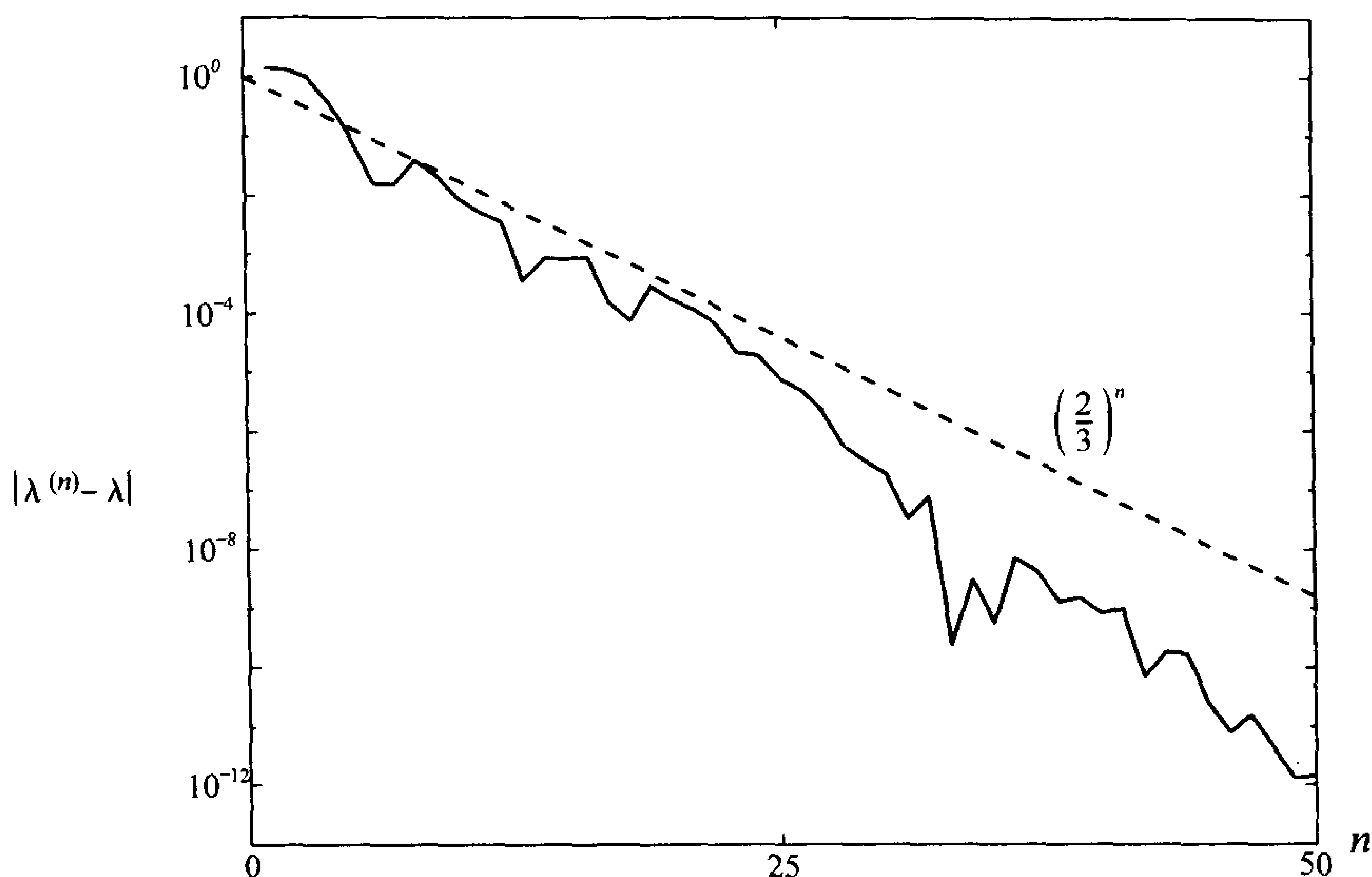


图 34-4 最右阿诺尔迪特征值估计的收敛性

这个性质的解释如下. 为极小化 (34.3), 在 A 的每一特征值上, p^n 必须取一个粗略的极小值. 例如, 考虑候选多项式 $p(z)=z^{n-1}(z-\tilde{\lambda})$, 其中 $\tilde{\lambda}$ 是接近 λ 的某个数. 在单位圆盘内的 A 的每个特征值上, $|p(z)|$是 1 阶的或更小, 然而, 在 $z=\lambda$, 它有数量

$$|p(\lambda)|\approx\left(\frac{3}{2}\right)^n|\tilde{\lambda}-\lambda|$$

(如果 λ 恰好等于$\frac{3}{2}$, 那么这是一个等式). 当 n 很大时, $\left(\frac{3}{2}\right)^n$ 非常巨大. 对于这个数也是 1 阶的, 为平衡它, $|\lambda-\tilde{\lambda}|$必须取得足够小, 即如在 (34.6), 为阶$\left(\frac{2}{3}\right)^n$.

另一个特点在图 34-4 中很明显. 在最初几十次迭代之后, 收敛开始加速, 这个现象在阿诺尔迪子空间迭代中是普遍的. 这里的情况是, 求解单位圆盘附近某些 A 的其余外特征值的迭代开始实施. 如果 A 的维数是 $m=300$, 那么特征值充分稠密, 充满了单位圆盘, 但如图 34-4 所指出的, 在 50 步迭代内这样的加速没有明显出现.

264

习 题

34.1 给出 $A\in\mathbb{C}^{m\times m}$, $b\in\mathbb{C}^m$和 $p\in P^n$, 假设要计算 $p(A)b$. 开始的自然步骤是用 Horner 规则, 它可以写成.

$$p(z)=c_0+z(c_1+z(c_2+\cdots+z(c_{n-1}+z)\cdots))\tag{34.7}$$

(a) 写出一个基于 (34.7) 的 **for** 循环 (在纸上，不是在计算机上) 来计算 $p(\boldsymbol{A})$，然后把这个结果应用到 $\boldsymbol{b}$ 上. 试确定用此算法直到首阶所需要的 flop 数.

(b) 写出计算 $p(\boldsymbol{A})\boldsymbol{b}$ 的另外一个 **for** 循环，好得多的一个算法，其中 $\boldsymbol{b}$ 在开始就引入到方法中. 再次确定使用这一算法到首阶所需要的 flop 数.

(c) 在初等的数值分析教科书中，在多项式计算中推荐使用 Horner 规则，其原因是，它比计算幂 z^k，用系数 c_k 乘和相加这样的明显的方法要快. 试证明，相反地，计算$p(\boldsymbol{A})$或 $p(\boldsymbol{A})\boldsymbol{b}$，Horner 规则比明显的方法没有快太多.

34.2 已经考察了阿诺尔迪多项式 p^n 极小化$\|p^n(\boldsymbol{A})\boldsymbol{b}\|$. 可以给出关于特征值更为清楚信息的另一多项式是*理想的阿诺尔迪多项式* (ideal Arnoldi polynomial)，也称为 $\boldsymbol{A}$ 的 *切比雪夫多项式* (Chebyshev polynomial)，它定义为，惟一的 $p^* \in P^n$ 极小化$\|p^n(\boldsymbol{A})\|$(由于没有快的方法来计算这个多项式，所以在实际中这个多项式是不使用的).

(a) 证明 p^* 是存在的.

(b) 证明，若 $p^*(\boldsymbol{A}) \neq 0$，那么 p^* 是惟一的. [提示：假定 p_1 和 p_2 是关于给定 A 和 n 的不同的理想阿诺尔迪多项式. 令 $p = \frac{1}{2}(p_1 + p_2)$，并考虑对应于极大奇异值的 $p(\boldsymbol{A})$的奇异向量. 这是一道难题.]

34.3 设 $\boldsymbol{A}$ 是 $N \times N$ 的双对角线矩阵，其元素 $a_{k,k+1} = a_{k,k} = k^{-\frac{1}{2}}$，$N = 64$. (在极限 $N \to \infty$，$\boldsymbol{A}$ 变为非自伴紧算子.)

(a) 画出显示谱 $\Lambda(\boldsymbol{A})$的曲线和显示对于$\epsilon = 10^{-1}, 10^{-2}, 10^{-3}$和 10^{-4}的ϵ-伪谱 $\Lambda_\epsilon(\boldsymbol{A})$的边界曲线 (习题 26.1 和 26.2).

(b) 从随机初始向量开始，在步 $n = 1, 2, \cdots, 30$ 运行阿诺尔迪迭代并计算里茨值，画出指明收敛到 $\boldsymbol{A}$ 的特征值的收敛速率曲线，并对结果作出说明.

(c) 阿诺尔迪迭代可以用 $\boldsymbol{H}_n$ 或 $\tilde{\boldsymbol{H}}_n$的伪谱来逼近 $\boldsymbol{A}$ 的伪谱. [在后面的情况中，$\Lambda_\epsilon(\tilde{\boldsymbol{H}}_n)$的边界是用条件 $\sigma_{\min}(z\boldsymbol{I} - \tilde{\boldsymbol{H}}_n) = \epsilon$，或等价地用$\|(z\boldsymbol{I} - \tilde{\boldsymbol{H}}_n)^+\| = \epsilon^{-1}$来确定的，其中 $\boldsymbol{I}$ 是单位矩阵的矩形等价形式.] 用画出 $n = 5, 10, 15, 20$ 的 $\tilde{\boldsymbol{H}}_n$的ϵ-伪谱来试验这一思想，它们与 $\boldsymbol{A}$ 的相应的伪谱有多接近?

265

第 35 讲 GMRES

在上一讲中，指出了如何用阿诺尔迪方法求特征值．在本讲中，将指出它也可用于求解方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$．此类的标准算法称为 GMRES，是广义极小剩余（generalized minimal residuals）的简称．

35.1 在 $\mathcal{K}_n$ 中剩余极小化

如上两讲一样，设 $\boldsymbol{A}\in\mathbb{C}^{m\times m}$ 是方阵，$\boldsymbol{b}\in\mathbb{C}^{m}$ 是向量，以及 $\mathcal{K}_n$ 表示（33.5）的克雷洛夫子空间 $\langle\boldsymbol{b},\boldsymbol{A}\boldsymbol{b},\cdots,\boldsymbol{A}^{n-1}\boldsymbol{b}\rangle$．然而，现在的目的是解方程组 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$，所以还假定 $\boldsymbol{A}$ 是非奇异的．对于这个问题的精确解用记号：$\boldsymbol{x}_*=\boldsymbol{A}^{-1}\boldsymbol{b}$ 来表示是方便的．

GMRES 的思想是一个极小化．在 n 步，用使剩余 $\boldsymbol{r}_n=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_n$ 的范数减小到最小的向量 $\boldsymbol{x}_n\in\mathcal{K}_n$ 来逼近 $\boldsymbol{x}_*$．换言之，用解由图 35-1 说明的最小二乘问题来确定 $\boldsymbol{x}_n$．

266

解这个最小二乘问题的明显方法如下．

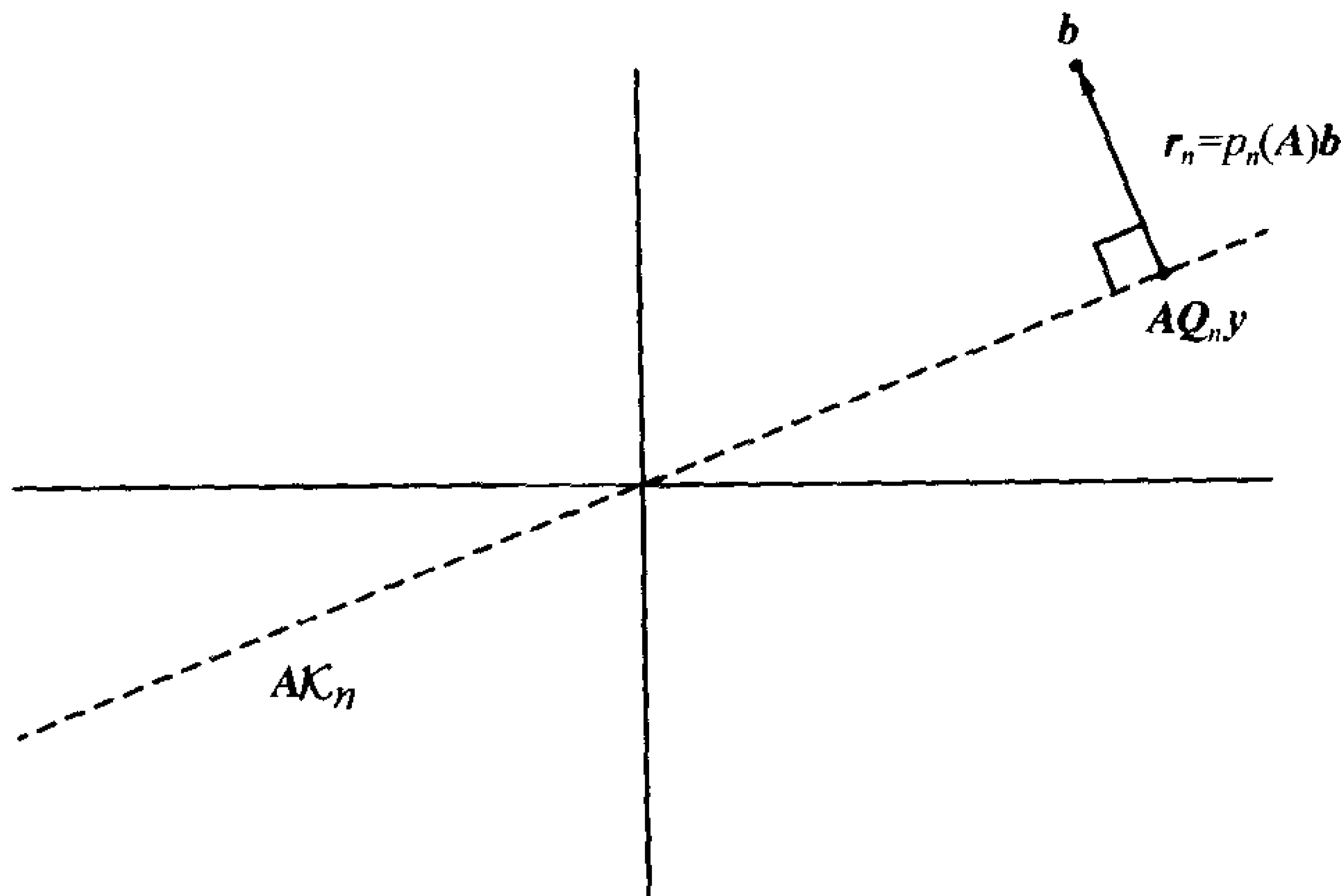


图 35-1 基于 GMRES 的最小二乘多项式逼近问题：极小化剩余范数 $\|r_n\|$．与图 34-1 作比较

令 $\boldsymbol{K}_n$ 是 $m\times n$ 的克雷洛夫矩阵（33.6），所以有

$$\boldsymbol{A}\boldsymbol{K}_n=\left[\begin{array}{c|c|c|c} \boldsymbol{A}\boldsymbol{b} & \boldsymbol{A}^2\boldsymbol{b} & \cdots & \boldsymbol{A}^n\boldsymbol{b} \end{array}\right]. \tag{35.1}$$

这个矩阵的列空间是 $A\ \mathcal{K}_n$. 于是,问题为求一个向量 $c\in\mathbb{C}^n$,使得

$$\|AK_nc-b\| = \text{极小值}. \tag{35.2}$$

其中 $\|\cdot\|=\|\cdot\|_2$，如在本书的这一部分经常使用的一样. 这可以用类似于（33.7）的 $\boldsymbol{AK}_n$ 的 QR 因子分解来达到. 一旦 c 求出，那么令 $\boldsymbol{x}_n=\boldsymbol{K}_n\boldsymbol{c}$.

然而，刚才描述的过程在数值上是不稳定的，并且这一过程构造了并不需要的因子 $\boldsymbol{R}$. 实际上是用下面方法来代替上述做法. 用阿诺尔迪迭代（算法 33.1）构造克雷洛夫矩阵 $\boldsymbol{Q}_n$ 的一个序列. $\boldsymbol{Q}_n$ 的列向量 $\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots$ 张成了逐次的克雷洛夫子空间 $\mathcal{K}_n$. 于是用 $\boldsymbol{x}_n=\boldsymbol{Q}_n\boldsymbol{y}$ 代替了 $\boldsymbol{x}_n=\boldsymbol{K}_n\boldsymbol{c}$. 代替（35.2），最小二乘问题是求一向量 $y\in\mathbb{C}^n$，使得

$$\|\boldsymbol{A}\boldsymbol{Q}_n\boldsymbol{y}-\boldsymbol{b}\| = \text{极小值}. \tag{35.3}$$

表面上，问题（35.3）有 $m\times n$ 维，然而，在实际中，由于克雷洛夫子空间的特殊构造，其维数基本上是 $(n+1)\times n$. 可以作如下说明来揭示这一事实. 应用（33.3）把这个方程变换到

$$\|\boldsymbol{Q}_{n+1}\tilde{\boldsymbol{H}}_n\boldsymbol{y}-\boldsymbol{b}\| = \text{极小值}. \tag{35.4}$$

267

现在在范数内部的两个向量均在 $\boldsymbol{Q}_{n+1}$ 的列空间中，因此用 $\boldsymbol{Q}_{n+1}^*$ 左乘不改变范数. 于是，另外一个等价问题是

$$\|\tilde{\boldsymbol{H}}_n\boldsymbol{y}-\boldsymbol{Q}_{n+1}^*\boldsymbol{b}\| = \text{极小值}. \tag{35.5}$$

最后，通过克雷洛夫矩阵 $\{\boldsymbol{Q}_n\}$ 的构造可以注意到，$\boldsymbol{Q}_{n+1}^*\boldsymbol{b}$ 等于 $\|\boldsymbol{b}\|\boldsymbol{e}_1$，其中照例 $\boldsymbol{e}_1=(1,0,0,\cdots)^*$. 于是，最终达到 GMRES 最小二乘问题的最后形式：

$$\|\tilde{\boldsymbol{H}}_n\boldsymbol{y}-\|\boldsymbol{b}\|\boldsymbol{e}_1\| = \text{极小值}. \tag{35.6}$$

在 GMRES 的 n 步，对 $\boldsymbol{y}$ 解这个问题，然后令 $\boldsymbol{x}_n=\boldsymbol{Q}_n\boldsymbol{y}$.

35.2 GMRES 的技巧

下面将完成 GMRES 算法的推导. 一个高水平的描述如下.

算法 35.1 GMRES

$q_1=b/\|b\|$

for $n=1,2,3\ldots$

〈阿诺尔迪迭代的 n 步，算法 33.1〉

求 y 最小化 $\|\tilde{H}_ny-\|b\|e_1\|$ （$=\|r_n\|$）

$x_n=Q_ny$.

在每一步，GMRES 对于所有向量 $\boldsymbol{x}_n \in \mathcal{K}_n$ 极小化剩余 $\boldsymbol{r}_n = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n$ 的范数. 量 $\|\boldsymbol{r}_n\|$ 是在求 $\boldsymbol{y}$ 中计算的，并不需要从 $\boldsymbol{x}_n$ 来显式地计算.

循环的内部，算法 35.1 的“求 $\boldsymbol{y}$”步是一个具有海森伯格结构的 $(n+1)\times n$ 矩阵最小二乘问题. 它可以用在第 11 讲中描述的通常方式的 QR 因子分解来求解，由于海森伯格结构，所以计算量为 $O(n^2)$ 的 flop. 此外，可以用更为特殊化的方法，进一步节省其工作量，不用对系列矩阵 $\tilde{\boldsymbol{H}}_1, \tilde{\boldsymbol{H}}_2, \cdots$，独立地构造 QR 因子分解，而可以用更新步骤从 $\tilde{\boldsymbol{H}}_{n-1}$ 的 QR 因子分解来得到 $\tilde{\boldsymbol{H}}_n$ 的 QR 因子分解. 所有这些所需要的是单个 Givens 旋转（习题 10.4）和 $O(n)$ 工作量，见习题 35.4.

35.3 GMRES 和多项式逼近

在上一讲中，已经指出了用阿诺尔迪迭代计算特征值与逼近问题（34.3）有关：求 $p^n \in P^n$ 使其极小化 $\|p^n(\boldsymbol{A})\boldsymbol{b}\|$，其中 P^n 表示 n 次首 1 多项式的集合. GMRES 迭代也是解一个逼近问题，惟一的差别是现在的多项式空间为 [268]

$$P_n = \{\text{满足 } p(0) = 1 \text{ 的次数} \leqslant n \text{ 的多项式 } p\}. \tag{35.7}$$

利用多项式系数表示的，现在正规化是 $c_0 = 1$ 而不是 $c_n = 1$. 表示法上，从上指标变为下指标作为从在 $z = \infty$ 的正规化变为在 $z = 0$ 的正规化的一个提示.

下面说明如何能把 GMRES 约化为在 P_n 中的多项式逼近. 迭代 x_n 可以写为

$$\boldsymbol{x}_n = q_n(\boldsymbol{A})\boldsymbol{b}. \tag{35.8}$$

其中 q 是 $n-1$ 次多项式；其系数是（35.2）的向量 c 的元素，相应的剩余 $\boldsymbol{r}_n = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_n$ 是 $\boldsymbol{r}_n = (\boldsymbol{I} - \boldsymbol{A}q_n(\boldsymbol{A}))\boldsymbol{b}$，其中 p_n 是用 $p_n(z) = 1 - zq(z)$ 来定义的多项式. 换言之，对于某个多项式 $p_n \in P_n$，有

$$\boldsymbol{r}_n = P_n(\boldsymbol{A})\boldsymbol{b} \tag{35.9}$$

GMRES 方法选择 p_n 的系数使其极小化这个剩余的范数. 于是，我们已经指出，GMRES 对于 $n = 1, 2, 3, \cdots$ 逐次地解下面的逼近问题.

> **GMRES 逼近问题** 求 $p_n \in P_n$，使得
>
> $$\|p_n(\boldsymbol{A})\boldsymbol{b}\| = \text{极小值}. \tag{35.10}$$

如阿诺尔迪迭代一样，GMRES 满足某些不变性性质. 下面的定理从（35.10）容易证明.

定理 35.1 令 GMRES 迭代应用到矩阵 $\boldsymbol{A} \in \mathbb{C}^{m\times m}$，方法如上面所描述.

标度不变性. 如果对于某一 $\sigma \in \mathbb{C}$，$\boldsymbol{A}$ 变换到 $\sigma\boldsymbol{A}$，$\boldsymbol{b}$ 变换到 $\sigma\boldsymbol{b}$，那么剩余 $\{\boldsymbol{r}_n\}$ 变换到 $\{\sigma\boldsymbol{r}_n\}$.

酉相似变换下的不变性. 如果对某一酉矩阵 U,A 变换到 UAU^*，并且 b 变换到 Ub，那么剩余$\{r_n\}$变换到$\{U^*r_n\}$.

由于正规化 $p(0)=1$ 包含了平移－依赖点 0. 所以 GMRES 在平移下没有不变性. 相反，其性质强有力地依赖于原点的位置——不严格地说，依赖于 A 的条件数. 269

35.4　GMRES 的收敛性

GMRES 收敛有多快？在$\|r_n\|/\|b\|$减少到一个满意的水平，例如 10^{-3}或 10^{-16}之前，必须迭代多少步 n？作为一个实际问题，这常常变成设计一个好的预处理器的问题（第 40 讲）. 在数学上，这个问题是研究 A 的什么性质决定了$\|r_n\|$的大小.

我们以两个观察作为开始. 第一个是 GMRES 是单调收敛的：

$$\|r_{n+1}\| \leqslant \|r_n\|. \tag{35.11}$$

其原因是对于子空间 $\mathcal{K}_n$，$\|r_n\|$要尽可能小. 通过把 $\mathcal{K}_n$ 扩大到空间 $\mathcal{K}_{n+1}$，只能减小剩余范数，在最坏的情况下，剩余范数保持不变. （注意 $P_n \subseteq P_{n+1}$，而 $P^n \not\subseteq P^{n+1}$.）第二个是最多 m 步后这方法必须收敛，至少在忽略舍入误差情况：

$$\|r_m\| = 0. \tag{35.12}$$

对于一般数据 A 和 b，由于 $\mathcal{K}_m = \mathbb{C}^m$，所以这种收敛将发生，在特殊情况下，如果对于某一 $n<m$，x_* 碰巧在于 $\mathcal{K}_n$ 中，那么收敛将更早发生. 作为 GMRES 的数学的剩余，方程（35.12）是有用的；注意到若要 GMRES 迭代有用，那么它必须在 $n \ll m$ 步收敛到满意的精度，所以方程（35.12）没有什么实用价值.

为了得到关于收敛性的更多有用信息，必须转向多项式逼近问题（35.10）. 已知$\|r_n\| = \|p_n(A)b\| \leqslant \|p_n(A)\|\|b\|$是极小的，除了右端项 b 相对于 A 有特殊的结构的问题，确定这个量大小的关键因素通常是$\|p_n(A)\|$. 即确定 GMRES 的收敛速率的一般是不等式

$$\frac{\|r_n\|}{\|b\|} \leqslant \inf_{p_n \in P_n} \|p_n(A)\|. \tag{35.13}$$

这带给我们一个数学上很优雅的问题：给定矩阵 A 和数 n，$\|p_n(A)\|$能有多小？这个问题是解方程组的克雷洛夫子空间迭代法收敛性的几乎所有分析的基础.

35.5　在谱上尽可能小的多项式

给定矩阵 A 和数 n，$\|p_n(A)\|$能达到多小？得到此估计的标准方法是求满足 $p(0)=1$的多项式 $p(z)$，使其在谱 $\Lambda(A)$上尽可能的小. 如果 p 是一个多项式并且 S 是复平面内的一个集合，用

$$\|p\|_S = \sup_{z \in S} |p(z)|. \tag{35.14}$$

270

来定义标量$\|p\|S$. 设A是可对角化的，满足，对于某一非奇异矩阵V和对角矩阵Λ有$A=V\Lambda V^{-1}$，那么有

$$\|p(A)\| \leq \|V\|\|p(\Lambda)\|\|V^{-1}\| = \kappa(V)\|p\|_{\Lambda(A)} \tag{35.15}$$

把这个结果与（35.13）结合起来，就给出了下面关于GMRES收敛性的基本定理.

定理 35.2 在GMRES迭代的n步，剩余r_n满足

$$\frac{\|r_n\|}{\|b\|} \leq \inf_{p_n\in P_n}\|p_n(A)\| \leq \kappa(V)\inf_{p_n\in P_n}\|p_n\|_{\Lambda(A)}, \tag{35.16}$$

其中$\Lambda(A)$是A的特征值的集合，V是A的特征向量的非奇异矩阵（假设A是可对角化的），以及$\|p_n\|_{\Lambda(A)}$是由（35.14）定义的.

这个定理可以用文字总结如下：如果A离正则不太远（所谓正则的是指$\kappa(V)$不太大），并且如果固有的正规化n次多项式能被发现它的大小，在谱$\Lambda(A)$上随n很快减小，那么GMRES迅速地收敛.

例 35.1 这里是一个数值例子，设A是200×200矩阵，其元素是来自于均值为2，标准差为$0.5/\sqrt{200}$的实正态分布的独立样本. 在MATLAB中，

```
m = 200;  A = 2*eye(m)+0.5*randn(m)/sqrt(m)       (35.17)
```

图35-2指出了A的特征值，点集在半径为$\frac{1}{2}$，中心在$z=2$的圆盘内近似地均匀分布（习题12.3）. 图35-3指出了把GMRES迭代应用到问题$Ax=b$上，其中$b=(1,1,\cdots,1)^*$的收敛曲线. 在此情况下，收敛是非常平稳的，速率近似为4^{-n}，这个原因不难看出. 由于A的谱近似地充满了已指明的圆盘，所以$\|p(A)\|$是由选择$p(z)=\left(1-\frac{z}{2}\right)^n$来决定的近似极小化. 因为$I-A/2$是改变了比例的随机矩阵，所以它的谱充满了半径为$\frac{1}{4}$原点为$O$的圆盘，因此有$\|p(A)\|=\|(I-A/2)^n\|\approx4^{-n}$. 这个矩阵$A$是良态的，其条件数$\kappa(A)\approx2.03$. 来自于正规性的偏差是适中的，其$\kappa(V)\approx141$.

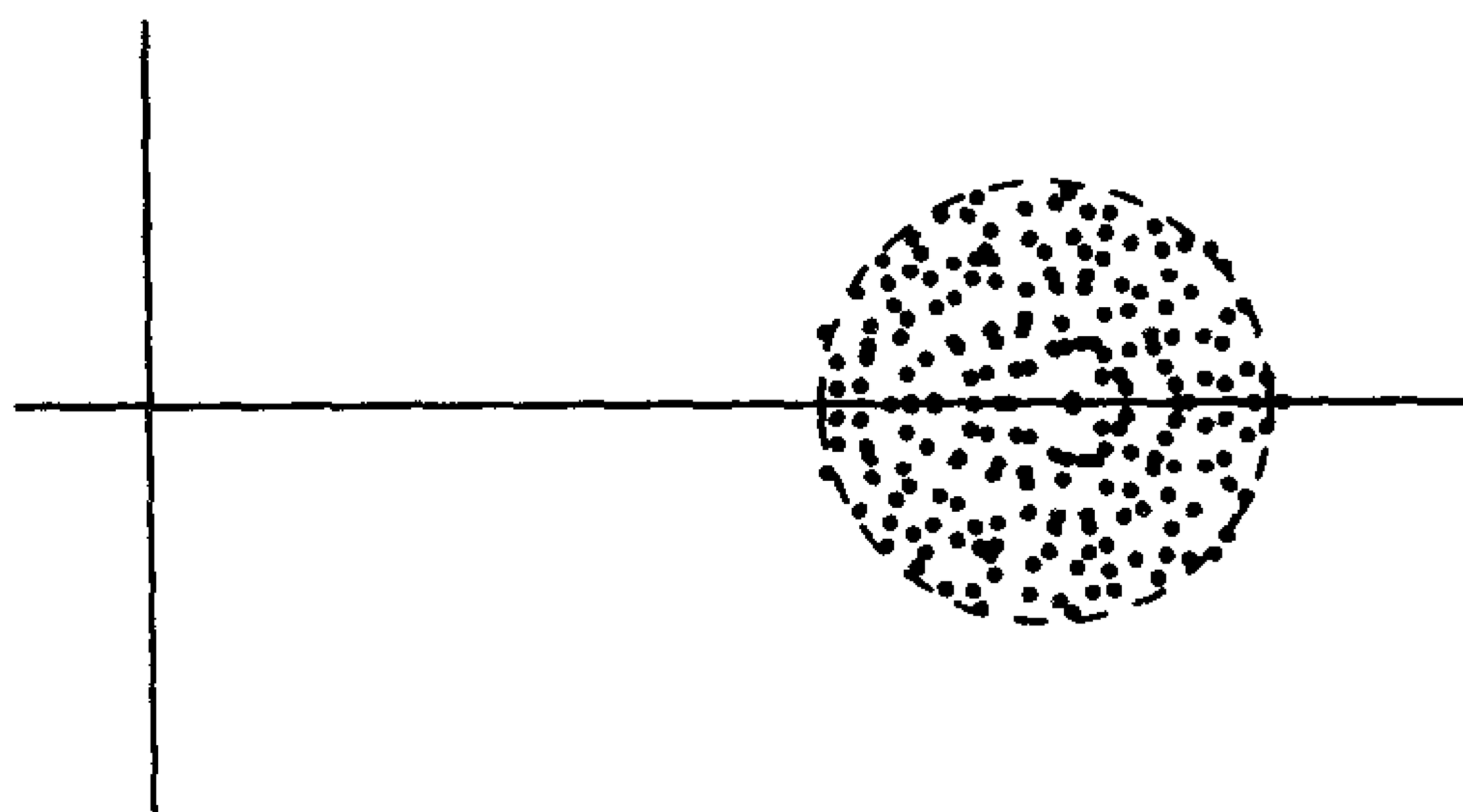


图35-2 （35.17）的200×200矩阵A的特征值. 虚曲线是在$\mathbb{C}$中，中心$z=2$，半径为$\frac{1}{2}$的圆. 特征值在这个圆盘内近似地均匀分布

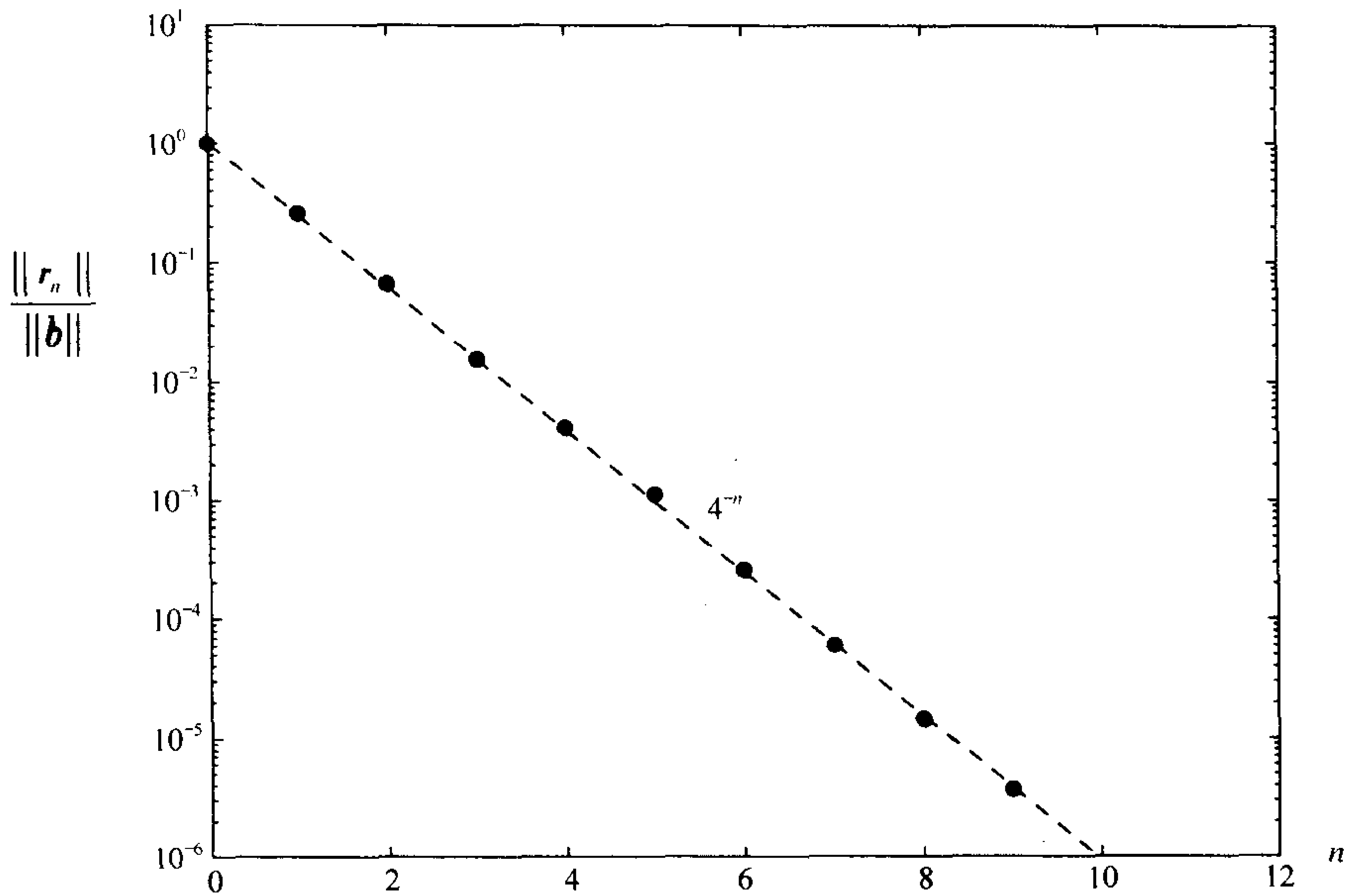


图 35-3　对于同一矩阵 A 的 GMRES 收敛曲线. 当 A 是有良好行为（或有良好预处理）的矩阵时，快速、平稳的收敛是在理想情况下克雷洛夫子空间迭代的说明

271 ≀ 272

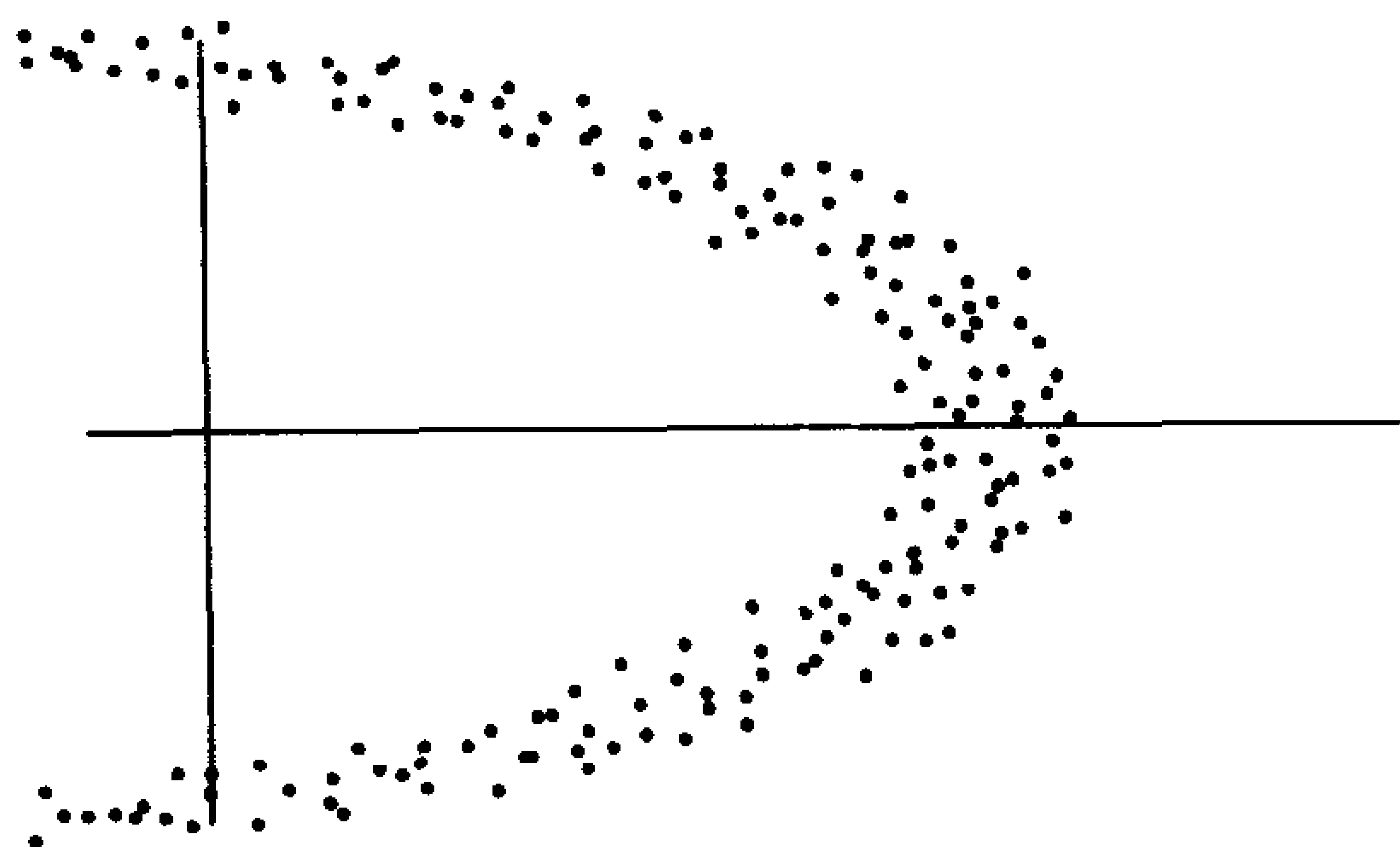
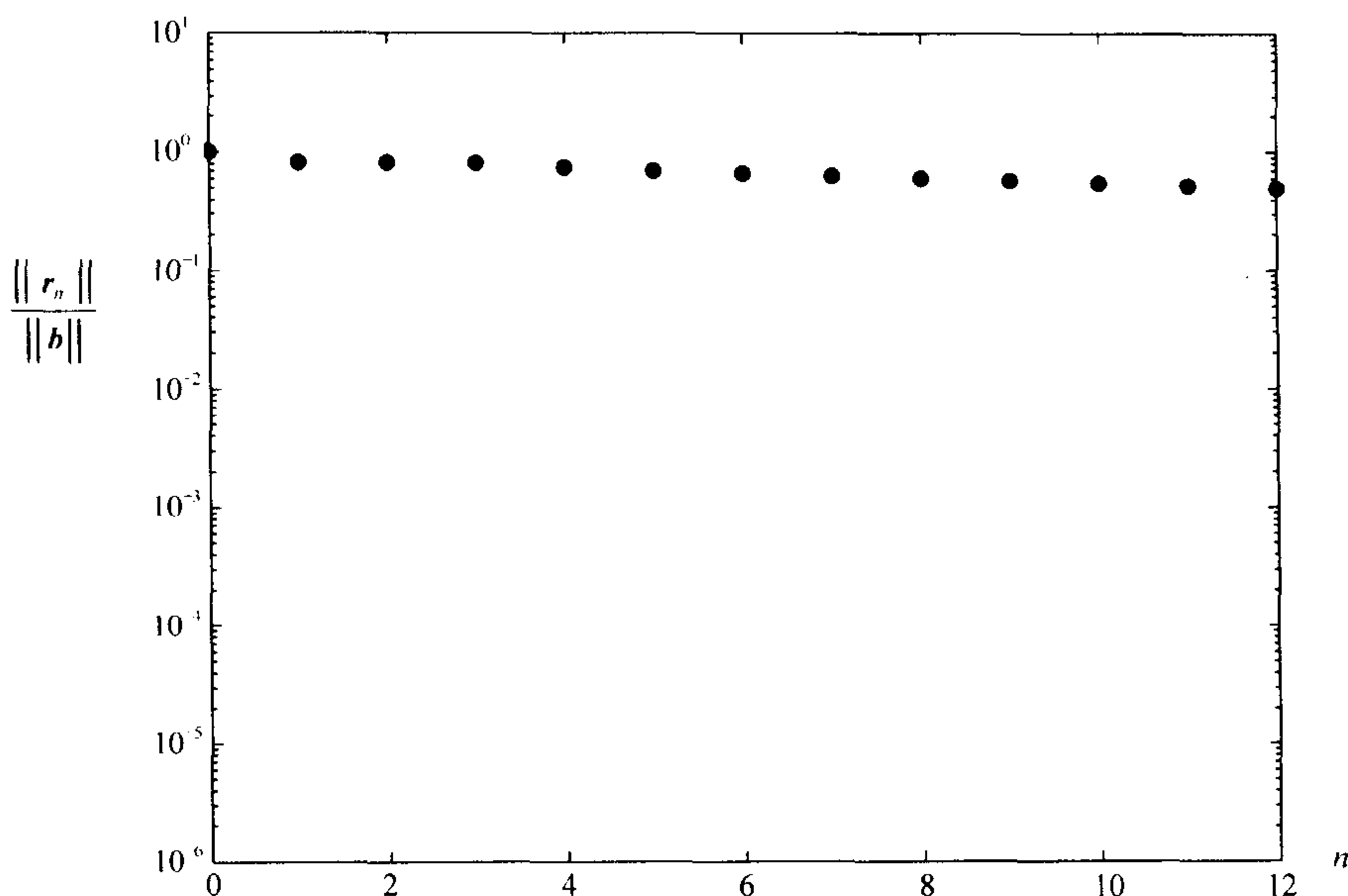


图 35-4　200 × 200 矩阵的特征值，其矩阵如（35. 17）的但修改了对角线的矩阵. 现特征值在一边围绕原点



273

图 35-5 图 35-4 的矩阵的 GMRES 的收敛曲线. 收敛已大大地减慢了. 当迭代方法如这样停滞时，应是寻找更好预处理器的时候

图 35-3 说明在顺利情况——当矩阵 A 有良好行为（通常意味着应用了好的预处理器）之下矩阵迭代的收敛. 可以看到，在 10 次迭代之后达到了 6 位数字的精确度，由于在每一步上，其工作量是由矩阵-向量乘法占优势，所以其代价近似地为 $10\times 2m^2=8.0\times 10^5$flop. 用高斯消元法解同样的方程组需要$\frac{2}{3}m^3\approx 5.3\times 10^6$flop. 即使没有稀疏性的好处以及 $m=200$ 不是很高，但这个改进近似地达到了 7 倍. 对于 $m=2000$的同样例子，GMRES 要胜过高斯消元法 70 多倍. 对于具有同样谱性质但有 90% 或 99% 稀疏性的 2000 × 2000 矩阵，分别改进为 700 倍或 7000 倍，并且GMRES 所需的存储也显著地减少. □

例 35.2 如果矩阵的特征值在另一面"围绕原点"，那么不能期望如此快速的收敛。图 35-4 ~ 图 35-5 提供了一个例子. 现在矩阵是 $A'=A+D$，其中 A 为（35.17）的矩阵，D 为对角矩阵，其复元素为

$$d_k=(-2+2\sin\theta_k)+i\cos\theta_k,\quad \theta_k=\frac{k\pi}{m-1},0\leqslant k\leqslant m-1.$$

这在图 35-4 中是显然的，现位于半圆暗影中的特征值围绕原点弯曲. 对于这个问题，迭代计算和高斯消元法一样，收敛速率比以前更差. 现在条件数 $\kappa(A)\approx 4.32$ 和 $\kappa(V)\approx 54.0$，所以在收敛中的恶化不能仅用条件数来解释；导致困难的是特征值的位置，而不是它们的数量（或奇异值的数量）. 如果弧围绕谱更进一步扩展，那么收

敛将更坏.　□

习　题

35.1　证明，如果 $S\subseteq\mathbb{C}$ 包含了无限多个点，那么（35.14）在全部复系数多项式的向量空间上定义了一个范数. 说明，如果 S 仅有有限多个点，那么结论是错误的.

35.2　(a) 令 $S\subseteq\mathbb{C}$ 是一个集合，其凸包包含 0，而 0 在 S 内部. 即 S 不包含在与原点不相交的半平面内. 证明，不存在 $p\in P_1$（即没有满足 $p(0)=1$ 的一次多项式 p）使得 $\|p\|_S<1$.

(b) 令 $\boldsymbol{A}$ 是矩阵，未必为正规矩阵，其谱 $\Lambda(\boldsymbol{A})$ 满足性质（a）. 证明，不存在 $p\in P_1$ 使得 $\|p(\boldsymbol{A})\|<1$.

(c) 虽然在图 35-5 中的收敛是慢的，显然有 $\|\boldsymbol{r}_1\|<\|\boldsymbol{r}_0\|$，说明为什么这个结论与（b）的结果不矛盾. 说明 GMRES 或许可能达到 $\|\boldsymbol{r}_1\|<\|\boldsymbol{r}_0\|$ 的哪类多项式 $p_1\in P_1$.

35.3　递推关系 $\boldsymbol{x}_{n+1}=\boldsymbol{x}_n+\alpha\boldsymbol{r}_n=x_n+\alpha(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_n)$ 称为理查森迭代（Richardson iteration）其中 α 为标量常数.

(a) 在 n 步，这对应什么多项式 $p(\boldsymbol{A})$?　[274]

(b) 对于图 35-2 的矩阵 $\boldsymbol{A}$，可取的 α 选择是什么? 所期望的相应的收敛速度是多少?

(c) 对于图 35-4 的矩阵提出相同的问题.

35.4　(a) 对于解算法 35.1 的最小二乘问题，用吉文斯旋转（习题 10.4）写出基于 QR 因子分解的 $O(n^2)$ 算法.

(b) 如果问题的 $n-1$ 步已经求得，如原书 p268（见页边标注）所述，说明如何将运算计数改进到 $O(n)$.

35.5　GMRES 算法（算法 35.1）叙述以初始估计 $\boldsymbol{x}_0=0$，$\boldsymbol{r}_0=\boldsymbol{b}$ 为开始.（同样应用到 CG 和 BCG，算法 38.1 和 39.1）. 说明，如果希望用任意的初始估计 $\boldsymbol{x}_0$ 作为开始，那么可以用关于右边的 $\boldsymbol{b}$ 稍作修正来完成.

35.6　对于更大的 n，GMRES 在运算和存储的代价是过高的. 在这种情况中，称为 k 步重新开始的 GMRES 或 GMRES(k) 的方法经常被使用. 此方法是，在 k 步之后，GMRES 迭代是用当前的向量 x_k 作为初始估计重新开始.

(a) 对于固定的 k 及增加的 n，比较 GMRES 和 GMRES(k) 所需要的大致的运算计数和存储.

(b) 举出一个例子，在此例中期望 GMRES(k) 用几乎如 GMRES 一样少的迭代步数就收敛（因此在运算计数中少很多）.

(c) 再举出一个例子，在此例中 GMRES(k) 不收敛，而 GMRES 收敛.　[275]

第 36 讲 兰乔斯迭代

在前面 3 讲中讨论了非埃米尔特矩阵问题的克雷洛夫子空间迭代，在第 39 讲中将回到非埃米尔特问题，在那里将不局限于阿诺尔迪和 GMRES. 然而，在本讲和下面两讲中，将限定在埃米尔特情形，并作重要的简化.

36.1 三项递推

兰乔斯迭代是限定在 $\boldsymbol{A}$ 是埃米尔特的情形下的阿诺尔迪迭代. 为简化记号，进而假定，在此讲和下面两讲中，$\boldsymbol{A}$ 是实的并且是对称的.

让我们考虑在这特殊情形中，阿诺尔迪方法将发生什么. 当然，第 33 讲和第 34 讲的所有等式仍将适用，并在每一公式中用^{T} 来代替*. 注意到的第一件事情是，从 (33.12) 可以得到的里茨矩阵 $\boldsymbol{H}_n$ 是对称的. 因此它的特征值、里茨值或兰乔斯估计 (33.10) 都是实的. 由于 $\boldsymbol{A}$ 的特征值是实的，所以这似乎是很自然的.

注意到的第二件事情是更惊人的，因为 $\boldsymbol{H}_n$ 既是对称的又是海森伯格，所以它是三对角的. 这表示在阿诺尔迪迭代（算法 33.1）的内循环中，限制 1 到 n 可以用 $n-1$到 n 来代替. 于是，兰乔斯迭代恰好包含三项递推式而非（$n+1$）项递推式 (33.4). 其结果果，兰乔斯迭代的每一步比阿诺尔迪迭代对应的步廉价的多. 在第 38 讲中将看到，对于解 $\boldsymbol{Ax}=\boldsymbol{b}$，类似地共轭梯度迭代的每一步比 GMRES 对应的步廉价的多.

276

$\boldsymbol{H}_n$ 是三对角的这个事实如此重要，以致于考察它是怎样从 $\boldsymbol{A}$ 的对称性中产生的是有价值的. 关键等式是 (33.12)，对于实矩阵 $\boldsymbol{A},\boldsymbol{H}_n$ 和 $\boldsymbol{Q}_n$ 可以把等式写成元素方式

$$h_{ij} = \boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{q}_j \tag{36.1}$$

由于 $\boldsymbol{Aq}_j\in\langle \boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_{j+1}\rangle$和克雷洛夫向量是正交的，所以此等式意味着对于 $i>j+1$，有 $h_{ij}=0$. 取其转置有

$$h_{ij} = \boldsymbol{q}_j^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{q}_i. \tag{36.2}$$

如果 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$，那么用如前同样的证论，对于 $j>j+1$ 有 $h_{ij}=0$. 导致三项递推关系的简单论证适用于任何自伴算子，而不仅适用于矩阵.

36.2 兰乔斯迭代

由于对称的三对角矩阵仅包含两个相异的向量，所以通常用新的变量代替一般

记号 a_{ij}. 令 $\alpha_n = h_{nn}$ 和 $\beta_n = h_{n+1,n} = h_{n,n+1}$，那么 $\boldsymbol{H}_n$ 变为

$$\boldsymbol{T}_n = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix} \tag{36.3}$$

用此记号，算法 33.1 采取下面的形式.

算法 36.1　兰乔斯迭代

$\beta_0 = 0, q_0 = 0$, 任给 $b, q_1 = b/\|b\|$
for $n = 1, 2, 3, \cdots$
　　$v = Aq_n$
　　$\alpha_n = q_n^{\mathrm{T}} v$
　　$v = v - \beta_{n-1} q_{n-1} - \alpha_n q_n$
　　$\beta_n = \|v\|$
　　$q_{n+1} = v/\beta_n$

277

每步由一个矩阵与向量乘法、一个内积和一对向量运算所组成，如果 $\boldsymbol{A}$ 具有足够的稀疏性或具有可以廉价计算的矩阵与向量乘积的其他结构，那么这样的迭代将没有太多困难地应用到维数为几万或几十万的问题上.

下面的定理总结了兰乔斯迭代的一些性质（当然，当执行准确的算术运算时与在本书中所有这样的定理一样）. 这里并无新意；这些是对阿诺尔迪迭代的定理 33.1 和 34.1 的结果在新记号下的重新叙述.

定理 36.1　由兰乔斯迭代产生的向量 q_n 组成的矩阵 $\boldsymbol{Q}_n$ 是克雷洛夫矩阵（33.6）的约化 QR 因子，

$$\boldsymbol{K}_n = \boldsymbol{Q}_n \boldsymbol{R}_n. \tag{36.4}$$

三对角矩阵 $\boldsymbol{T}_n$ 是相应的投影

$$\boldsymbol{T}_n = \boldsymbol{Q}_n^* \boldsymbol{A} \boldsymbol{Q}_n, \tag{36.5}$$

并且逐次迭代与下面的公式有关，

$$\boldsymbol{A}\boldsymbol{Q}_n = \boldsymbol{Q}_{n+1} \tilde{\boldsymbol{T}}_n, \tag{36.6}$$

此式可以写成在 n 步的三项递推形式，

$$\boldsymbol{A}\boldsymbol{q}_n = \beta_{n-1}\boldsymbol{q}_{n-1} + \alpha_n \boldsymbol{q}_n + \beta_n \boldsymbol{q}_{n+1}. \tag{36.7}$$

只要兰乔斯迭代不中断（即 $\boldsymbol{K}_n$ 有满秩 n），T_n 的特征多项式是解阿诺尔迪/兰乔斯逼

近问题（34.3），即达到

$$\|p^n(A)b\| = \text{极小值} \tag{36.8}$$

的惟一多项式 $p^n \in P^n$.

36.3　兰乔斯和电荷分布

在实际中，正如阿诺尔迪迭代用于非对称矩阵（第34讲）一样，兰乔斯迭代用于计算大的对称矩阵的特征值. 在每步 n，或在偶然出现的步上，增长的三对角矩阵 $\boldsymbol{T}_n$ 的特征值是用标准方法来确定的，对于给定的矩阵 $\boldsymbol{A}$ 和初始向量 $\boldsymbol{q}_1$，这些是里茨值或“兰乔斯估计”（33.10）. 通常看到这些数值中会有一些几何收敛到某些极限值，它们就是预期的 $\boldsymbol{A}$ 的特征值.

如同用阿诺尔迪迭代一样，通常最先得到的是 $\boldsymbol{A}$ 的分离特征值. 这个结论可以 278 用下面的经验法则做得更精确.

> 如果 $\boldsymbol{A}$ 的特征值比切比雪夫点更均匀地分布，
> 那么兰乔斯迭代将倾向于找到分离特征值.

下面是这个叙述所包含的内容. 假设 $\boldsymbol{A}$ 的 m 个特征值 $\{\lambda_j\}$ 在实轴上的一个区间附近合理地稠密地分布，由于兰乔斯迭代是标度不变性和平移不变性（定理34.2），所以不失一般性，可设这个区间是 $[-1,1]$. 在 $[-1,1]$ 内的 m 个切比雪夫点由下面公式来定义

$$x_j = \cos\theta_j, \quad \theta_j = \frac{(j-\frac{1}{2})\pi}{m}, \quad 1 \leq j \leq m. \tag{36.9}$$

精确的定义并不重要，值得关注的事情是这些点在靠近端点时平方聚集，在区间内部点之间间隔为 $O(m^{-1})$，而靠近 ± 1 时为 $O(m^{-2})$. 经验法则断言，如果 $\boldsymbol{A}$ 的特征值 $\{\lambda_j\}$ 比这些点更均匀地分布——在端点有更少的聚集——那么用兰乔斯迭代计算的里茨值将趋于首先收敛到分离特征值. 特别地，近似地把均匀特征值分布将导致快速收敛到分离特征值. 相反地，如果 $\boldsymbol{A}$ 的特征值在端点比平方聚集更多——在实际中不是如此普遍——那么可以期望收敛到某些“内围特征值”.

这些观察可以给出物理上的解释. 考虑 m 个点电荷自由地在区间 $[-1,1]$ 周围移动，假设位于 x_j 和 x_k 上的电荷之间的排斥力与 $|x_j - x_k|^{-1}$ 成比例.（对于在3D中的电子电荷，其力为 $|x_j - x_k|^{-2}$，但在2D中这就变为 $|x_j - x_k|^{-1}$，其中把每一点可以看为在3D中的直线与平面的交.）设这些电荷它们自身分布于在 $[-1,1]$ 内的一个极小-能量平衡中. 因此这极小-能量分布与切比雪夫分布近似地相同，在极限 $m\to\infty$ 情况下，它们两者均收敛到与 $(1-x^2)^{-\frac{1}{2}}$ 成比例的极限连续电荷密度分布.

把 $\boldsymbol{A}$ 的特征值考虑为点电荷，如果它们在一个区间内近似地分布于极小-能量结

构中，那么兰乔斯迭代将是无用的. 在 $n=m$ 步之前，它们将不收敛. 然而，如果分布与这个有很大不同，则可能会很快收敛到某些特征值，即特征值在有“太小电荷”的区域内，所谓“太小电荷”的含义为，如果点是自由移动的，那么更多的电荷将在这里聚集. 现在经验法则可以再叙述为：

在关于平衡分布的“太小电荷”区域内，兰乔斯迭代有助于收敛到特征值.

这个观察的解释依赖于兰乔斯迭代与多项式逼近的联系（36.8）. 一些细节在习题 36.2 中作出.

279

36.4 例　　子

兰乔斯迭代的收敛最好用数值例子来说明，设 A 是 203×203 矩阵

$$A=\mathrm{diag}(0,.01,.02,\cdots,1.99,2,\quad 2.5,\quad 3.0).\qquad(36.10)$$

A 的谱由在 [0,2] 上密集的特征值的集合和两个分离特征值，2.5 和 3.0 组成. 我们以随机初始向量 q_1 为开始来完成兰乔斯迭代.

图 36-1 显示了在 $n=9$ 步的里茨值和相联系的兰乔斯多项式 7 个里茨值位于 [0,2] 内并且在这个区间上多项式是一致地小；可以看出，初始阶段的里茨值倾向于聚集在端点附近. 另外两个里茨值位于 2.5 和 3.0 的特征值附近. 为首的 3 个里茨值是

$$1.93,\ 2.48,\ 2.999962.$$

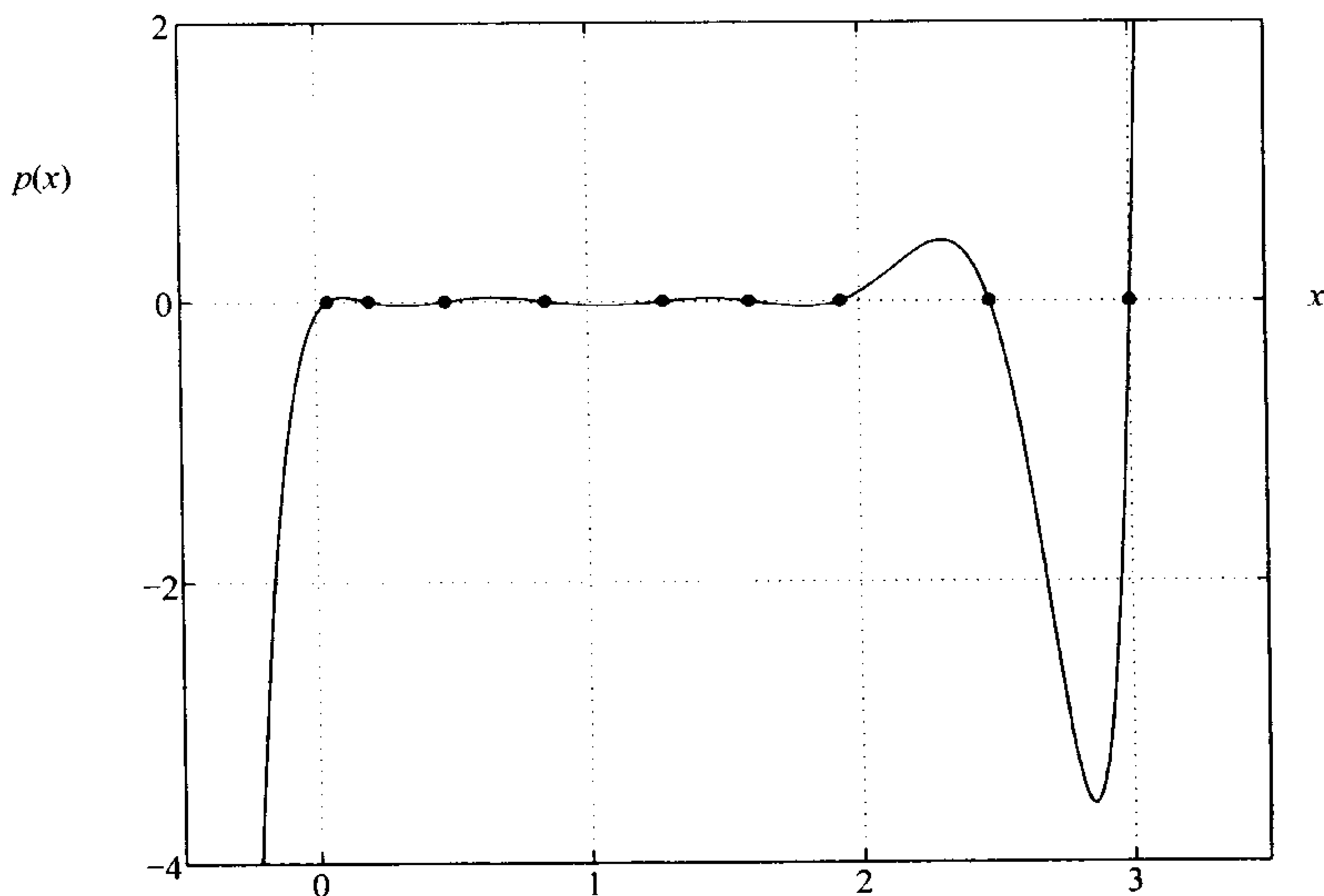


图 36-1 矩阵（36.10）的兰乔斯迭代第 9 步上的兰乔斯多项式曲线. 根是里茨值或“兰乔斯特征值估计”. 在整个集合[0,2]∪{2.5}∪{3.0}上多项式是小的. 为达到这些，必须有一个根位于 2.5 附近，另一个根更接近于 3

显然，较低的特征值有小的精度，但首个特征值有5位数字的精度. 如这样情况的曲线给出了一个思想，为什么分离特征值有助于精确地估计. 对于$x\approx 3$, $p(x)$的图形是如此陡峭，以致于如果$p(3)$是小的，那么p的根是非常靠近的了. 图形的这个陡峭性与这点附近出现的“太小电荷”有关. 如果电荷在 $[0,3]$ 上自由移动使能量减少到最小，那么有更多的点将聚集在$x=3$附近，并且在那里$p(x)$不会如此陡峭.

在第20步，为首的3个里茨值是

$$1.9906,\ 2.499999999987,\ 3.00000000000000.$$

对于首特征值有15位数字的精度，第2特征值有12位数字的精度. $p(x)$的曲线在靠近点2.5和3.0处有相应地陡峭，注意到，第3个特征值的收敛也开始出现，这个事实反映了在 $[0,2]$ 中的特征值是均匀地分布而不是切比雪夫分布.

收敛的“空中摄影照片”出现在图36-2中，此图指出了从$n=1$到$n=20$所有步的里茨值. 此曲线的每个纵向片断对应于在一次迭代上的里茨值；连接点的线帮助眼睛跟踪所发生的现象但没有精确的含义. 曲线指出，大约$n=5$以后显著地收敛到首特征值，在大约$n=10$显著收敛到其次的特征值. 在包含其他特征值的区间 $[0,2]$ 内，曲线显示了里茨值的密度近似地正比于$(1-x^2)^{-1/2}$，在端点有很清晰的聚束.

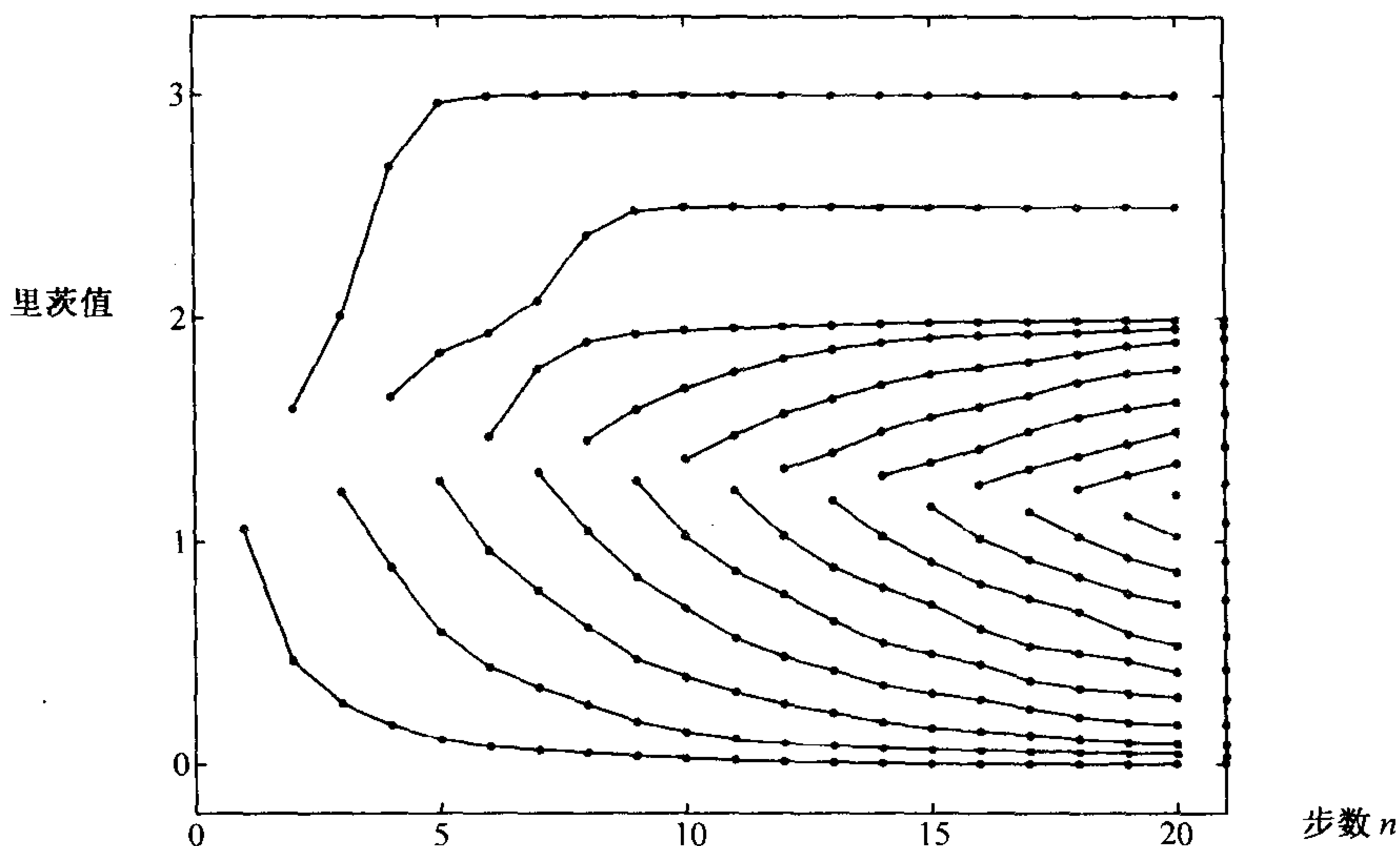


图36-2　应用到同一矩阵的兰乔斯迭代最初20步的里茨值. 特征值2.5和3.0的收敛是几何的. 对于出现在$[0,2]$中的部分谱的单个特征值收敛效果较差. 然而，$[0,2]$中的里茨值逼近该区间中的切比雪夫点，这些点已在右边界上用点标出

36.5 舍入误差和“重像（Ghost）”特征值

舍入误差对于兰乔斯迭代有复杂的影响，事实上，对于基于三项递推关系的数值线性代数的所有迭代都如此，容易发现这个困难的来源，在基于 n-项递推关系的迭代中，例如阿诺尔迪或 GMRES，向量 $q_1, q_2, q_3, \cdots$ 用显式格拉姆-施密特运算来实施正交. 然而，如兰乔斯和共轭梯度的三项递推依赖于由数学恒等式“自动地”产生的向量 $\{q_j\}$ 的正交性. 在实际中，在有舍入误差的情况下，这样的恒等式不再精确地保持，并且在许多次迭代之后，失掉了正交性.

在实际的兰乔斯迭代中失去正交性似乎是特别不好的，但情形比这更难以捉摸. 碰巧，失去正交性与里茨值收敛到 A 的特征值有密切联系. 关于这个问题，尽管没有人们希望的那么多了解，但也有了很多了解；在此不作详细讨论.

由于这些复杂性，所以在本书中定义下的关于兰乔斯和共轭梯度迭代的稳定性，还没有简单明了的定理. 尽管如此，在实际中这些迭代是非常有用的. 图 36-3 给出了这个方法的思想，在此方法中不稳定性经常表明，在实际中迭代是有用的，此图是图 36-2 的重复，但用 120 步迭代代替了 20 步迭代. 当特征值 3.0 的第 2 个拷贝出现在里茨值中时，此时约为 30 步. 在此之前一切显得如期望的那样. 第 3 个拷贝大约在 60 步出现，第 4 个拷贝大约在 90 步出现等等. 同时，特征值 2.5 的附加拷贝也在大约 40 步、80 步和（刚开始就可看出的）120 步出现. 这些附加的里茨值称为“重像”特征值，并且它们与 A 的对应特征值的实际重数毫无关系. 282

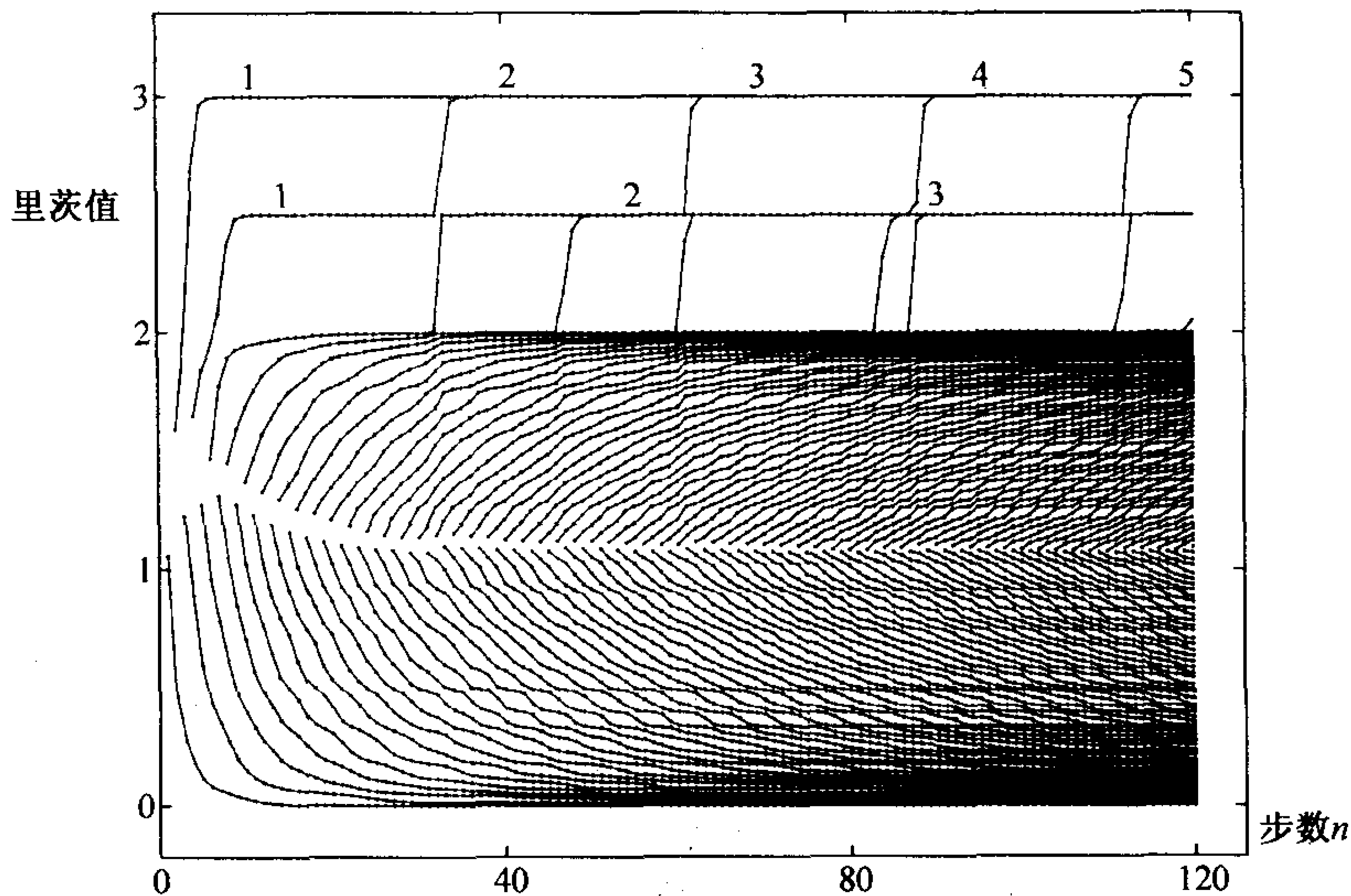


图 36-3 继续到兰乔斯迭代的 120 步. 数字指明了里茨值的重数. 注意出现了特征值 3.0 的 4 个“重像”拷贝和特征值 2.5 的 2 个“重像”拷贝

对重像特征值现象的严格分析是复杂的，然而，直观解释是不难作出的. 一个思想是，在出现舍入误差情况下，应该考虑到，$\boldsymbol{A}$ 的每个特征值不是作为一个点而是作为大小大致地为 $O(\epsilon_{\text{机器}}\|\boldsymbol{A}\|)$ 的一个小区间. 重像特征值是由需要 $p(z)$ 不仅在精确的特征值上，而且在整个这些小区间上是小的这一事实而出现的. 另外，有些不同的解释是，里茨值收敛到 $\boldsymbol{A}$ 的一个特征值消去了在参与运算的向量中对应的特征向量的分量；但在舍入误差出现时，随机噪声必须被期望再次稍微激发出那一分量. 在充分多次迭代之后，这个以前消去的分量已得到足够的放大，以致于需要用另一个里茨值再次消去它——然后再放大，再消去.

两个解释抓住了在浮点运算中兰乔斯迭代的性质的一些实质，或许第 2 个解释有更多数量上的精确性.

习　题

36.1　在第 27 讲中已经指出，对称矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times m}$ 的特征值是瑞利商 $r(\boldsymbol{x})=(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x})/(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x})$，$\boldsymbol{x}\in\mathbb{R}^{m}$ 的平稳值. 证明在兰乔斯迭代的 $\boldsymbol{n}$ 步，当 $\boldsymbol{x}$ 限于 $\mathcal{K}_n$ 时里茨值是 $r(\boldsymbol{x})$ 的平稳值.

36.2　考虑多项式 $p\in P^n$，即对某些 $z_k\in\mathbb{C}$，$p(z)=\prod_{k=1}^{n}(z-z_k)$.

(a) 把 $\log|p(z)|$ 写成对应于点 z_k 的 n 项的和.

(b) 如果电荷以反比排斥来释出，那么说明为什么包含 z_k 的项可以解释为对应于在 z_k 的负单位点电荷的势. 于是，$\log|p(z)|$ 可以看为由 n 点电荷导出的在 z 上的位势.

(c) 用 $-1/n$ 来取代每一电荷 -1 并取极限 $n\to\infty$，得到积分为 -1 的连续电荷密度分布 $\mu(\zeta)$，可以期望此积分与当 $n\to\infty$ 时多项式 $p\in P^n$ 的零点的极限密度有关. 写出相应于 $\mu(\zeta)$ 的一个积分来表示位势 $\varphi(z)$，并说明它与 $|p(z)|$ 的联系.

283

(d) 令 S 为 $\mathbb{C}$ 中的闭的、有界子集且没有孤立点. 假定要求支集在 S 中的分布 $\mu(z)$，使得 $\max_{z\in S}\varphi(z)$ 极小化. 给出一个为什么这样的 $\mu(z)$ 应在整个 S 上满足 $\varphi(z)=$ 常数的理由（不严格的）. 说明为什么这表示“电荷”处于平衡状态，没有受到净力. 换言之，S 是像一个 2D 电的导体，在此导体上电荷的量 -1 有自由分布. 除了一个附加的常数，$\varphi(z)$ 是关于 S 的格林函数（Green function）.

(e) 作为说明 p. 279（英文原文）的经验法则的一步，假设 $\boldsymbol{A}$ 是实对称矩阵，其谱在 $[a,b]\cup\{c\}\cup[d,e]$，$a<b<c<d<e$，中稠密地分布. 于是对于集合 $S=[a,e]$，(b,d) 是一个“太小电荷”的区域. 说明为什么可以期望一个里茨值快速收敛到 c，以及根据与集合 $S'=[a,b]\cup[c,d]$ 相结合的平衡位势 $\varphi(z)$ 来估计收敛速度.

36.3　设 $\boldsymbol{A}$ 为 1000×1000，对称矩阵，其元素除了在对角线上 $a_{ij}=\sqrt{i}$，在上、下次对角线上 $a_{ij}=1$，以及在第 100 条上，下次对角线，即 $|i-j|=100$，上 $a_{ij}=1$，以外均为零. 用兰乔斯迭代确定 $\boldsymbol{A}$ 的最小特征值到 6 位数字的精度.

36.4　作为第 34 讲的阿诺尔迪双纽线的特殊情况，“兰乔斯双纽线”能够用来说明兰乔斯迭代的收敛性. 寻求一个方法来修改习题 36.3 的程序以便画出在每步上的兰乔斯双纽线. 此方法不必是优美的、效率高的或在数值上稳健的. 对于图 36-2 的例子和自选的一个例子在 $n=1,2,\cdots,12$ 步上画出兰乔斯双纽线的图.

284

第 37 讲　由兰乔斯到高斯求积

如果离散的向量变成在 [$-1,1$] 上的连续函数，且矩阵 $\boldsymbol{A}$ 取为用 $\boldsymbol{x}$ 逐点乘法的算子，那么兰乔斯迭代变成了通过三项递推关系构造正交多项式的标准过程. 由此高斯求积公式是简短的一步，其节点和权可以由解对称三对角矩阵的特征值问题来计算.

37.1　正交多项式

在第 7 讲中，考虑了 QR 因子分解的连续模拟. 现在考虑兰乔斯迭代的连续模拟，如在上一讲中，兰乔斯迭代把注意力放在实向量（现在为函数）和实对称矩阵（现在为线性算子）上.

要做的第一件事情是，用在 [$-1,1$] 上的实值函数的向量空间 $L^2[-1,1]$ 来代替 $\mathbb{R}^m$. 两个函数 $u,v\in L^2[-1,1]$ 的内积定义为

$$(\boldsymbol{u},\boldsymbol{v}) = \int_{-1}^{1} \boldsymbol{u}(x)\boldsymbol{v}(x)\mathrm{d}x, \tag{37.1}$$

函数 $u\in L^2[-1,1]$的范数是$\|u\|=(u,u)^{\frac{1}{2}}$.

在第 7 讲中，取 $\boldsymbol{A}$ 为 “$[-1,1]\times n$ 矩阵”，其列是 x 的幂. 这里，对于兰乔斯迭代. $\boldsymbol{A}$ 应是方阵而不是长方阵. 对应于用 x 逐点乘法，将取 $\boldsymbol{A}$ 为 “$[-1,1]\times[-1,1]$矩阵”，即 $\boldsymbol{A}$ 是一个线性算子，对于每一个 $u\in L^2[-1,1]$ 用方程

285

$$(\boldsymbol{A}u)(x) = xu(x) \tag{37.2}$$

来定义这个线性算子，这个算子类似于对角矩阵，沿“对角线”值 $x\in[-1,1]$ 是连续取值的（把 $\boldsymbol{A}$ 描述为有核 $k(x,y)=x\delta(x-y)$的积分算子可使对角矩阵思想精确，其中 δ 为狄拉克 δ 函数.）特别地. $\boldsymbol{A}$ 是对称的，并且由于对称性，阿诺尔迪过程特殊化为兰乔斯过程.

还有一条要限定. 假定初始函数 $b(x)$为非零常数，因此，对应的正规化初始函数 $q_1(x)$是 $q_1(x)=1/\sqrt{2}$.

现在兰乔斯迭代（算法 36.1）取为下面的形式.

算法 37.1　正交多项式的构造

$\beta_0=0, q_0(x)=0, q_1(x)=1/\sqrt{2}$

for $n=1,2,3,\cdots$

$v(x)=xq_n(x)$

$\alpha_n=(q_n,v)$

$v(x)=v(x)-\beta_{n-1}q_{n-1}(x)-\alpha_n q_n(x)$

$\beta_n=\|v\|$

$q_{n+1}(x)=v(x)/\beta_n$

注意到多项式 $q_1,q_2,q_3,\cdots$的次数为 $0,1,2,\cdots$，于是在这个序列中的 n 次多项式为 q_{n+1}.

算法 37.1 可以在许多书中找到，表面上与线性代数毫无关系. 算法 37.1 确实是构造在区间 $[-1,1]$ 上正交的勒让德多项式序列常用的三项递推关系. 然而，上述算法的叙述在记号和正规化方面是独特的。通常 n 次勒让德多项式写为 $P_n(x)$，n 次的并且用 $P_n(1)=1$ 来正规化. 而上面的 $q_{n+1}(x)$是通常 $P_n(x)$的数量倍数. 尽管如此，还称其为勒让德多项式.

勒让德多项式已在（7.11）和图 7-11 中讨论过了，它们是用上面提到的单项式的“$[-1,1]\times n$ 矩阵”的格拉姆-施密特因子分解来导出的. 那个矩阵具有由目前初始向量 $\boldsymbol{b}$ 和算子 $\boldsymbol{A}$ 生成的克雷洛夫矩阵（33.6）的形式，而这就是为什么在第 7 讲中的格拉姆-施密特过程会有与这里的兰乔斯过程同样的效果.

算法 37.1 称为“正交多项式的构造”而不是称为“勒让德多项式的构造”，其原因是前者更一般. 如果（37.1）用在被积函数中包含一个正的非常数权函数 $w(x)$ 来修改，那么得到了其他熟知的正交多项式，例如切比雪夫多项式和雅可比多项式. 这一讲的所有情况都可应用到这些更一般的族上，但我们不做详细讨论.

286

37.2　雅可比矩阵

前面几讲中关于阿诺尔迪和兰乔斯迭代的所有公式，对于刚才叙述的多项式正交化过程仍然是成立的. 当然，必须要作适当的说明，用$[-1,1]$来代替 $1,2,\cdots,m$ 且用（37.1）来代替通常的向量内积. 例如，现有“$[-1,1]\times n$ 矩阵”.

$$\boldsymbol{K}_n=\left[\begin{array}{c|c|c|c} 1 & x & \cdots & x^{n-1} \end{array}\right],\quad \boldsymbol{Q}_n=\left[\begin{array}{c|c|c|c} \boldsymbol{q}_1(x) & \boldsymbol{q}_2(x) & \cdots & \boldsymbol{q}_n(x) \end{array}\right], \tag{37.3}$$

并且它们与在（36.4）中一样恰好彼此有关.（正如刚才所述，$\boldsymbol{K}_n$ 是在第 7 讲中称为 $\boldsymbol{A}$ 的矩阵.）

在前讲中描述的三对角矩阵$\{T_n\}$是特别重要的. 由（36.5）和（36.6）给出的Q_n和A仍是$n \times n$离散矩阵，其元素是用（36.1）的模拟给出.

$$t_{ij} = (q_i(x), xq_j(x)) \tag{37.4}$$

在正交多项式的情况下，矩阵$\{T_n\}$称为雅可比矩阵. 三项递推关系（36.7）取为

$$xq_n(x) = \beta_{n-1}q_{n-1}(x) + \alpha_n q_n(x) + \beta_n q_{n+1}(x) \tag{37.5}$$

算法 37.1 的叙述或许在写法上易使人误解. 显然，在这个算法的每一步上包含了非平凡计算：确定α_n和β_n的内积(q_n, v)和范数$\|v\|$的计算. 如果（37.1）包含了一个任意的权函数$w(x)$，那么这些计算实际上是非平凡的. 然而，对于与勒让德多项式有关的特殊选择$w(x)=1$，以及也与其余经典的多项式族有关的各种选择，元素$\{\alpha_n\}$和$\{\beta_n\}$在解析上已知. 用（36.3）的符号，对于勒让德多项式有

$$\alpha_n = 0, \beta_n = \frac{1}{2}(1-(2n)^{-2})^{-\frac{1}{2}} \tag{37.6}$$

由于使用这些公式，算法 37.1 变成了一个平凡的机械的方法，一个三项递推关系而别无其他.

287

37.3　特征多项式

在正交多项式的这种情况下，阿诺尔迪/兰乔斯逼近问题（36.8）变成什么？这回答要求注意b和A的特殊选择，有$p(A)b = p(x)/\sqrt{2}$. 由此得出，(36.8）可以用下面的方式写出.

> **正交多项式逼近问题**. 求$p^n \in P^n$使得
>
> $$\|p^n(x)\| = \text{极小值}. \tag{37.7}$$

根据定理 36.1，其解是矩阵T_n的特征多项式.

由此，简短的一步就得到了惊人的结论. 注意到，任一$p \in P^n$可以写成形式$p(x) = Cq_{n+1}(x) + Q_n y$，其中$C$是一个常数——$q_{n+1}(x)$首项系数的倒数. 也注意到，由于函数$\{q_n(x)\}$是正交的，所以有$\|p\| = (C^2 + \|y\|^2)^{\frac{1}{2}}$. 由此得出在（37.7）中极小值是用取$y=0$来达到的，换言之，最多差一个常数$C$，$p^n(x)$与$q_{n+1}(x)$是一样的. 把这一结论表示成一个定理，定理的最后结论在习题 37.2 中证明.

定理 37.1　令$\{q_n(x)\}$是由算法 37.1 产生的正交多项式序列，令$\{T_n\}$是与三对角雅可比矩阵相联系的序列，并令p^n是T_n的特征多项式，那么对于$n=0,1,2,\cdots$

$$p^n(x) = C_n q_{n+1}(x), \tag{37.8}$$

其中 C_n 是常数. 特别地，$q_{n+1}(x)$ 的零点是 T_n 的特征值. 这几个零点是相异的并位于开区间（$-1,1$）内.

[这个定理对于计算非常重要. 为确定勒让德多项式的零点，人们必须做的是计算相关的雅可比矩阵的特征值，其矩阵元素由（37.6）以紧密的形式给出. 正如在前面几讲中所看到的，关于 $n\times n$ 对称三对角矩阵的特征值问题是良态的，并且可以很快地求解，仅需要 $O(n^2)$ 次 flop. 相反，从多项式的系数而不是从雅可比矩阵开始直接计算勒让德多项式的零点是效率低的，并且数值不稳定.]

37.4　求积公式

存在一个为什么勒让德多项式的零点在计算上令人感兴趣的原因：其零点是格拉姆-施密特求积公式的节点.

下面简单回顾一下数值求积的思想. 假设 $f(x)$ 是定义在［$-1,1$］上的函数，并且要计算积分

288

$$I(f)=\int_{-1}^{1}f(x)\,\mathrm{d}x \tag{37.9}$$

（如果感兴趣的积分区间不是［$-1,1$］，那么可以用变量的线性变换来处理.）考虑用有限和

$$I_n(f)=\sum_{j=1}^{n}w_jf(x_j) \tag{37.10}$$

来逼近 $I(f)$ 是自然的，其中有限和是用不依赖于 f 的选择的 n 个节点或横坐标 $x_j\in[-1,1]$ 的集合和相应的权 w_j 来确定的. 这是 n 个点的求积公式，由数值分析学者研究的概念可追溯到牛顿. 经常与自适应的误差估计，区间部分和阶的控制相结合的不同形式的这种公式是现在计算机上实现的大部分数值积分的基础.

节点的任一集合是求积公式的选择物. 下面的结果是范德蒙德矩阵的非奇异性的结论（习题 37.3）.

定理 37.2　设节点 $\{x_j\}$ 是［$-1,1$］内的 n 个相异点的任意集合，那么存在满足如下性质的权 $\{w_j\}$ 的惟一选择：求积公式（37.10）至少有 $n-1$ 阶精度，所谓 $n-1$ 阶精度是指，如果 $f(x)$ 是次数 $\leqslant n-1$ 的任一多项式，那么（37.10）是精确的.

如果节点 $\{x_j\}$ 从 -1 到 1 等距选取，由这个定理规定的求积公式称为牛顿-科茨公式，这是熟悉的梯形法则和辛普森法则在 n-点的推广. 牛顿-科茨公式有用这个定理保证的精度阶，但不能更高. 特别对于 n 较低的值，这些公式简单、有用. 对于较大的 n，它们的权 w_j 有振荡的符号并有阶为 2^n 的巨大振幅，由此引起数值不稳定.

37.5　高斯求积

高斯求积的思想不仅对权$\{w_j\}$，而且对节点$\{x_j\}$进行最优的选择，从而使得(37.10) 精度的阶尽可能提高. 恰巧，存在惟一的节点和权的选择达到这个要求，并由此得到的公式有 $2n-1$ 阶精度. 比起阶 $n-1$ 来，这是一个惊人的改进，对于光滑函数和函数计算值的固定数来说，精度数字的双倍数一般可以达到. 而且，甚至对于大的 n，使这些公式是稳定的，其权 w_j 都是正的.

289

双倍于 n 个节点求积公式的精度阶的节点 $x_1,\cdots,x_n$ 的魔集是什么？这正是勒让德多项式 $q_{n+1}(x)$的零点集合. 高斯或高斯-勒让德求积公式定义为定理 37.2 的求积公式 (37.10)，其中节点 $x_1,\cdots,x_n$ 为 $q_{n+1}(x)$的零点.

定理 37.3　*n-点高斯-勒让德求积公式恰好有 $2n-1$ 阶精度，并且求积公式(37.10) 没有高于这个精度的阶.*

证明　给出不同节点的任意集合$\{x_j\}$，令 $f(x)$为 $2n$ 次多项式 $\prod_{j=1}^{n}(x-x_j)^2$；那么有 $\boldsymbol{I}(f)>0$，但由于对每个节点 x_j 有 $f(x_j)=0$，所以 $\boldsymbol{I}_n(f)=0$. 于是，对于 $2n$ 次多项式，求积公式是不精确的.

另一方面，假设 $f(x)$是次数$\leqslant 2n-1$ 的任何一个多项式，并取$\{x_j\}$是高斯求积节点，$\{q_{n+1}(x)\}$的零点. 这个函数能因式分解成下面的形式

$$f(x)=g(x)q_{n+1}(x)+r(x),$$

其中 $g(x)$是次数$\leqslant n-1$ 的多项式，$r(x)$为余项，也是次数$\leqslant n-1$ 的多项式.（事实上，$r(x)$是关于 f 在点$\{x_j\}$的 $n-1$ 次插值多项式.）由于 $q_{n+1}(x)$与所有更低次数的多项式正交，所以有 $\boldsymbol{I}(gq_{n+1})=0$. 同时，由于对每个节点 x_j 有 $g(x_j)q_{n+1}(x_j)=0$，所以有 $\boldsymbol{I}_n(gq_{n+1})=0$. 因为 $\boldsymbol{I}$ 和 $\boldsymbol{I}_n$ 是线性算子，所以这些恒等式隐含着 $\boldsymbol{I}(f)=\boldsymbol{I}(r)$和 $\boldsymbol{I}_n(f)=\boldsymbol{I}_n(r)$. 但由于 $r(x)$的次数$\leqslant n-1$，所以由定理 37.2 有 $\boldsymbol{I}(r)=\boldsymbol{I}_n(r)$，组合这些结果就给出了所要求的 $\boldsymbol{I}(f)=\boldsymbol{I}_n(f)$. □

37.6　通过雅可比矩阵的高斯求积

本讲前半部分已经讨论了从兰乔斯迭代到勒让德多项式和从勒让德多项式到高斯求积. 甚至已经提供了一个快速和稳定算法去确定高斯求积公式的节点：刚产生的雅可比矩阵$\{\boldsymbol{T}_n\}$和计算它们的特征值.

用最后的观察结束这一叙述. 不但高斯求积公式的节点而且其权也能够从 $\boldsymbol{T}_n$ 的特征值问题中得到. 第 j 个高斯权原来简单地是 $w_j=2\ (v_j)_1^2$，即，$\boldsymbol{T}_n$ 的第 j 个特征向量的第 1 个分量的平方的 2 倍. 我们只叙述这个结果，不给出证明. 对于以一般权函数 $w(x)$用内积 (37.1) 确定的高斯求积公式的类似结果也成立.

290

定理 37.4　设 T_n 是由算法 37.1 或（36.5）定义的 $n\times n$ 雅可比矩阵（36.3），其元素 $\beta_1,\cdots,\beta_{n-1}$ 由（37.6）给出．令 $T_n=VDV^{\mathrm{T}}$ 是有 $V=[v_1|\cdots|v_n]$ 和 $D=\mathrm{diag}(\lambda_1,\cdots,\lambda_n)$ 的 T_n 的正交对角化．那么高斯-勒让德求积公式的节点和权由下面给出

$$x_j=\lambda_j,\quad w_j=2(v_j)_1^2,\quad j=1,\cdots,n. \tag{37.11}$$

37.7　例　子

作为积分光滑函数的高斯求积效率的说明，假定要计算积分

$$I(e^x)=\int_{-1}^{1}e^x\mathrm{d}x=2.35040239.$$

取 $n=4$，找到 4-点的高斯-勒让德求积的雅可比矩阵是

$$T_4=\begin{bmatrix}0 & 0.577350269 & & \\ 0.577350269 & 0 & 0.516397779 & \\ & 0.516397779 & 0 & 0.507092553\\ & & 0.507092553 & 0\end{bmatrix}$$

这个矩阵的特征值给出了节点

$$x_1=-x_4=0.861136312,\qquad x_2=-x_3=0.339981044,$$

以及特征向量的第一分量给出了相应的权

$$w_1=w_4=0.347854845,\qquad w_2=w_3=0.652145155.$$

计算和（37.10）给出

$$I_n(e^x)=2.35040209,$$

此与精确结果符合到大约 7 位数字．相比较，4 点牛顿-科茨公式给出 $I_n(e^x)\approx 2.3556$，仅精确到大约 3 位数字．

习　题

37.1　对于勒让德多项式，标准的递推关系是

291

$$P_n(x)=\frac{2n-1}{n}xP_{n-1}(x)-\frac{n-1}{n}P_{n-2}(x) \tag{37.12}$$

其中初始值 $P_0(x)=1,P_1(x)=x$.

(a) 验证（37.12）给出了（7.11）的多项式 $P_2(x)$ 和 $P_3(x)$.

(b) 由于 $\{P_n(x)\}$ 和 $\{q_{n+1}(x)\}$ 是不同的正规化，所以（37.12）与带系数（37.6）的递推关系（37.5）不同．写出对应于这些公式的两个三对角矩阵，并导出它们之间的联系．

(c) 用 (b) 的结果，确定 $q_{n+1}(x)$ 的公式，或等价地，确定 $\|P_n\|$ 的公式.

37.2 基于正交性的定义，证明 $q_{n+1}(x)$ 有 n 个不同的零点，它们都包含在开区间 $(-1,1)$ 内.（它们不同于从习题 25.1 得到的，这里是用直接方法证明.）

37.3 用次数小于等于 $n-1$ 的多项式在 n 个不同数据点 $\{x_j\}$ 上的 n 个数据值 $\{y_j\}$ 的插值问题，是在 (11.4) 中表示为方形的范德蒙德线性方程组.

(a) 用论述如果插值问题有解，那么它必须是惟一的这一论断，来证明范德蒙德矩阵是非奇异的.

(b) 写出在定理 37.2 未直接写明的类似方程组，用 (a) 的结果证明这个定理.

37.4 (a) 写出 (b) 行 MATLAB 程序，用于计算 n 个点的高斯-勒让德求积公式的节点和权，以及应用这些数去计算函数 f 的近似积分.

(b) 取 $f(x)=e^x$ 和 $n=4$，证实在正文中的例子. 此外，对于 $n=1,2,\cdots,40$ 在对数标度下画出 $|I(e^x)-I_n(e^x)|$ 的图形，并对结果作说明.

(c) 对函数 $f(x)=e^{|x|}$ 画出类似图形并作说明.

37.5 习题 37.4 的程序计算的勒让德多项式的零点，也称为在 $[-1,1]$ 内的**勒让德点**. 切比雪夫多项式的零点，**切比雪夫点**，是由显式公式 (36.9) 给出的. 在极限 $n\to\infty$ 的情况下，完成生成数一系列的计算以及画出尽可能精致的曲线，勒让德和切比雪夫点都趋于极限密度分布 $\mu(x)=\pi^{-1}(1-x^2)^{-\frac{1}{2}}$ (用习题 36.2 的符号). 为仔细研究下面的问题要进一步画出曲线和生成数：对于 n 的各种值，勒让德点和切比雪夫点靠得有多近？

292

第38讲　共轭梯度法

共轭梯度迭代是"原始的"克雷洛夫子空间迭代，是这些方法中最著名的一个也是科学计算的主流方法之一．共轭梯度迭代是在1952年由Hestenes和Stiefel发现的，在求解对称正定方程组时，如果特征值是良态分布的，那么它的求解非常快．

38.1　极小化剩余2-范数

如在前两讲中一样，设$A\in\mathbb{R}^{m\times m}$是实的和对称的，并且假定要求解非奇异方程组$\boldsymbol{Ax}=\boldsymbol{b}$，它具有精确解$\boldsymbol{x}_*=A^{-1}\boldsymbol{b}$．设$\mathcal{K}_n$表示由$\boldsymbol{b}$产生的第$n$个克雷洛夫子空间（33.5）．

$$\mathcal{K}_n=\langle\boldsymbol{b},A\boldsymbol{b},\cdots,A^{n-1}\boldsymbol{b}\rangle. \tag{38.1}$$

基于这个克雷洛夫子空间的一个方法是用GMRES解方程组．如在第35讲中所讨论的，这表示在n步，$\boldsymbol{x}_*$是由极小化$\|\boldsymbol{r}_n\|_2$的向量$\boldsymbol{x}_n$来逼近的，其中$\boldsymbol{r}_n=\boldsymbol{b}-A\boldsymbol{x}_n$．实际上，通常的GMRES算法比极小化$\|\boldsymbol{r}_n\|_2$所必要的工作量更多．因为$A$是对称的，所以基于在$n$步三项递推代替$(n+1)$-项递推的快速算法是适合的．其中之一名为共轭剩余或MINRES（"极小剩余"）．

这些方法，至少当它们应用到定型和不定型矩阵时，存在某些困难．在此不叙述它们而直接转到更简单和更重要的正定情形．

293

38.2　极小化误差的A-范数

假设A不仅是实的和对称的而且是*正定的*．正如在第23讲中所讨论的，这意味着A的特征值都是正的，或等价地，对每一非零$\boldsymbol{x}\in\mathbb{R}^m$有$\boldsymbol{x}^{\mathrm{T}}A\boldsymbol{x}>0$．在此假设下，由

$$\|\boldsymbol{x}\|_A=\sqrt{\boldsymbol{x}^{\mathrm{T}}A\boldsymbol{x}} \tag{38.2}$$

定义的函数$\|\cdot\|_A$是$\mathbb{R}^m$上的范数，这可以从定义（3.1）来证明，此范数称为A-范数．（如果W是A的楚列斯基因子或任何其他满足$W^{\mathrm{T}}W=A$的矩阵．那么A-范数与（3.3）的范数$\|\boldsymbol{x}\|_W$一样．）

采用A-范数的向量是$\boldsymbol{e}_n=\boldsymbol{x}_*-\boldsymbol{x}_n$，即在$n$步的误差．共轭梯度迭代可以描述如下：*共轭梯度迭代是产生惟一的具有在n步，$\|\boldsymbol{e}_n\|_A$达到极小这一性质的迭代序列$\{\boldsymbol{x}_n\in\mathcal{K}_n\}$的递推公式系统．*

下面将给出最初没有给出的 CG 迭代公式，并导出一些正交性质（定理 38.1）. 由此，关于 $\|e_n\|_A$ 的极小性质的论断作为推论（定理 38.2）得到，当把 CG 解释为非线性优化算法时，乃得到写出它的公式的动机.

38.3　共轭梯度迭代

下面是 Hestenes 和 Stiefel 给出的著名的迭代.

算法 38.1　共轭梯度（CG）迭代

$x_0 = 0, r_0 = b, p_0 = r_0$

for $n = 1, 2, 3, \cdots$

$\alpha_n = (r_{n-1}^{\mathrm{T}} r_{n-1}) / (p_{n-1}^{\mathrm{T}} A p_{n-1})$	步长
$x_n = x_{n-1} + \alpha_n p_{n-1}$	近似解
$r_n = r_{n-1} - \alpha_n A p_{n-1}$	剩余
$\beta_n = (r_n^{\mathrm{T}} r_n) / (r_{n-1}^{\mathrm{T}} r_{n-1})$	扩展此步
$p_n = r_n + \beta_n p_{n-1}$	搜索方向

在分析这些公式的数学性质之前，先在运算上来考察它们. 首先注意到 CG 迭代非常简单——用几行 MATLAB 可编成程序. 由于它仅处理 m-向量而不处理向量或矩阵的单个元素，所以它更简单，例如，比选主元的高斯消元法更简单. 惟一的复杂性（将不涉及）是选择收敛准则. 294

在每一步，CG 迭代包含了几个向量操作和一个矩阵-向量乘积，以及计算 Ap_{n-1}（在列表中有二次出现，但只需计算一次）. 如果 A 是稠密的且非结构的，那么矩阵-向量乘积在运算计数中占优势，每步为 $\sim 2m^2$ 次 flop. 如果 A 是稀疏的或有其他可利用的结构，那么 Ap_{n-1} 可以在如 $O(m)$ 运算一样少的情况下计算. 在此种情形中，每步的运算计数可以如 $O(m)$ 次 flop 一样少.

从定义算法的 5 行程序，可以推导下面的性质. 如在本书中所有定理中不明显提及舍入误差一样，这次也假定计算是在精确运算中实施的. 如果有舍入误差，那么这些性质不成立，并且解释仍给人深刻印象的 CG 的实施变成难以捉摸的事情.

定理 38.1　*设在对称正定矩阵问题 $\boldsymbol{Ax} = \boldsymbol{b}$ 上应用 CG 迭代（算法 38.1）. 只要迭代还不收敛（即 $\boldsymbol{r}_{n-1} \neq 0$），那么算法就没有用零作除数而继续进行，并且有下面的子空间恒等式：*

$$\begin{aligned}\mathcal{K}_n &= \langle \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n \rangle = \langle \boldsymbol{p}_0, \boldsymbol{p}_1, \cdots, \boldsymbol{p}_{n-1} \rangle \\ &= \langle \boldsymbol{r}_0, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_{n-1} \rangle = \langle \boldsymbol{b}, \boldsymbol{Ab}, \cdots, \boldsymbol{A}^{n-1}\boldsymbol{b} \rangle. \end{aligned} \tag{38.3}$$

而且，剩余是正交的，

$$r_n^{\mathrm{T}} r_j = 0 \qquad (j < n), \tag{38.4}$$

以及搜索方向是“**A**-共轭”，

$$p_n^{\mathrm{T}} A p_j = 0 \qquad (j < n). \tag{38.5}$$

证明　对 n 用归纳法证明；将非正规地写出证明概要. 从初始推测 $x_0 = 0$ 和公式 $x_n = x_{n-1} + \alpha_n p_{n-1}$，用归纳得出，$x_n$ 属于 $\langle p_0, p_1, \cdots, p_{n-1} \rangle$. 由 $p_n = r_n + \beta_n p_{n-1}$ 可以得出这与 $\langle r_0, r_1, \cdots, r_{n-1} \rangle$ 是一样的. 最后，从 $r_n = r_{n-1} - \alpha_n A p_{n-1}$ 可以得出这与 $\langle b, Ab, \cdots, A^{n-1} b \rangle$ 一样，由此（38.3）成立.

为证明（38.4），应用公式 $r_n = r_{n-1} - \alpha_n A p_{n-1}$ 和恒等式 $(A p_{n-1})^{\mathrm{T}} = p_{n-1}^{\mathrm{T}} A$ 来计算

$$r_n^{\mathrm{T}} r_j = r_{n-1}^{\mathrm{T}} r_j - \alpha_n p_{n-1}^{\mathrm{T}} A r_j.$$

如果 $j < n-1$，由归纳法，在右边的两项为零. 如果 $j = n-1$，倘若 $\alpha_n = (r_{n-1}^{\mathrm{T}} r_{n-1}) / (p_{n-1}^{\mathrm{T}} A r_{n-1})$. 那么右边的差是零. 现在除了用 $p_{n-1}^{\mathrm{T}} A r_{n-1}$ 来代替 $p_{n-1}^{\mathrm{T}} A p_{n-1}$ 之外，这与算法 38.1 的行 $\alpha_n = (r_{n-1}^{\mathrm{T}} r_{n-1}) / (p_{n-1}^{\mathrm{T}} A p_{n-1})$ 是一样的. 由于 p_{n-1} 与 r_{n-1} 的差为 $\beta_{n-1} p_{n-2}$，所以这个替代的效果是用 $\beta_{n-1} p_{n-1}^{\mathrm{T}} A p_{n-2}$ 来改变分母，由归纳假设，这是零. [295]

为证明（38.5），应用公式 $p_n = r_n + \beta_n p_{n-1}$ 来计算

$$p_n^{\mathrm{T}} A p_j = r_n^{\mathrm{T}} A p_j + \beta_n p_{n-1}^{\mathrm{T}} A p_j.$$

如果 $j < n-1$，由归纳法，在右边的两项再次为零（由于对于 n 的情况，（38.4）已经成立）. 如果 $j = n-1$，若 $\beta_n = -(r_n^{\mathrm{T}} A p_{n-1}) / (p_{n-1}^{\mathrm{T}} A p_{n-1})$，它可以写成等价形式 $\beta_n = (-\alpha_n r_n^{\mathrm{T}} A p_{n-1}) / (\alpha_n p_{n-1}^{\mathrm{T}} A p_{n-1})$，右边的和为零. 除了用 $r_n^{\mathrm{T}}(-\alpha_n A p_{n-1})$ 来代替 $r_n^{\mathrm{T}} r_n$ 和用 $p_{n-1}^{\mathrm{T}}(\alpha_n A p_{n-1})$ 来代替 $r_{n-1}^{\mathrm{T}} r_{n-1}$ 外，这与算法 38.1 的行 $\beta_n = (r_n^{\mathrm{T}} r_n) / (r_{n-1}^{\mathrm{T}} r_{n-1})$ 是一样的. 用归纳假设和算法 38.1 的第 3 行和第 5 行，这些代替再次证实了零效应. □

38.4　CG 的最优性

在推导正交性质（38.4）和（38.5）中，已经完成了有效的工作. 现在直接的事情是确认在每一步使 $\|e\|_A$ 极小化.

定理 38.2　设在对称正定矩阵问题 $Ax = b$ 上应用 CG 迭代. 如果迭代还没有收敛（即 $r_{n-1} \neq 0$），那么 x_n 是在 $\mathcal{K}_n$ 中极小化 $\|e_n\|_A$ 的惟一点. 收敛性是单调的，

$$\|e_n\|_A \leqslant \|e_{n-1}\|_A, \tag{38.6}$$

并且对某个 $n \leqslant m$，可以得到 $e_n = 0$.

证明　从定理 38.1 知道，x_n 属于 $\mathcal{K}_n$，为了证明它是在 $\mathcal{K}_n$ 中极小化 $\|e\|_A$ 的惟一点，考虑任一点 $x = x_n - \Delta x \in \mathcal{K}_n$，其误差 $e = x_* - x = e_n + \Delta x$. 计算

$$\|e\|_A^2 = (e_n + \Delta x)^{\mathrm{T}} A (e_n + \Delta x) = e_n^{\mathrm{T}} A e_n + (\Delta x)^{\mathrm{T}} A (\Delta x) + 2 e_n^{\mathrm{T}} A (\Delta x).$$

这个等式的最后项是 $2r_n^{\mathrm{T}}(\Delta x)$，即 r_n 与 $\mathcal{K}_n$ 中向量的内积，并且由定理 38.1，这样的内积是零. 这是使得 CG 迭代如此有用的关键的正交性质. 由此得到

$$\|e\|_A^2 = e_n^{\mathrm{T}} A e_n + (\Delta x)^{\mathrm{T}} A (\Delta x).$$

这些项中只有第 2 项依赖于 Δx，又因 A 是正定的，所以此项 $\geqslant 0$，并且取得零值的充分必要条件是 $\Delta x = 0$，即 $x_n = x$. 于是，$\|e\|_A$ 是极小的充分必要条件是 $x_n = x$，这正是所要求的.

296

定理余下的陈述现在容易得到了. 单调性质（38.6）是包含关系 $\mathcal{K}_n \subseteq \mathcal{K}_{n+1}$ 的结果，并且由于只要收敛还未达到，那么 $\mathcal{K}_n$ 就是 $\mathbb{R}^m$ 的 n 维子集，所以在至多 m 步必然达到收敛.

在浮点运算中，破坏了 CG 迭代至多 m 步收敛的保证. 在实际的计算机上，对于任意矩阵 A，当 $n = m$ 时，$\|e_n\|_A$ 的决定性的减少未必能完全观察到. 然而，在实际中，CG 不是用于任意矩阵的，而是用于一些矩阵，其谱，也许是由于预处理得到的谱，是具有对于 $n \ll m$ 可收敛到所希望的精度这样好的性能（第 32 讲）. 在 $n = m$，理论上的精确收敛与在科学计算中使用 CG 迭代没有关系. □

38.5 CG 作为最优化算法

刚才已经指出，CG 迭代有某一最优性质：它使得 $\|e_n\|_A$ 在 n 步于所有向量 $x \in \mathcal{K}_n$ 上极小化. 事实上，作为预测已经用了诸如“步长”和“搜索方向”这类术语，这个迭代可以解释为，求 $x \in \mathbb{R}^m$ 的非线函数极小值的标准形式的算法. 迭代的核心公式

$$x_n = x_{n-1} + \alpha_n p_{n-1}.$$

这是最优化中熟悉的等式，在此等式中，当前逼近 x_{n-1}，通过在方向 p_{n-1}（搜索方向）上移动距离 α_n（步长）来更新得到一个新的逼近 x_n. 用一系列这样的步骤，CG 迭代将试图求出非线性函数的极小值.

哪个函数？根据定理 38.2，这个回答应是 $\|e\|_A$，或等价地，$\|e\|_A^2$. 然而，虽然 $\|e\|_A^2$ 确实是 x 的函数，但若 x_* 未知，那它是不能求值的. 把 CG 解释为应用到不能求值的函数的一个优化过程不是很“标准的”!

另一方面，给定 A 和 b 以及 $x \in \mathbb{R}^m$，量

$$\varphi(x) = \frac{1}{2} x^{\mathrm{T}} A x - x^{\mathrm{T}} b \qquad (38.7)$$

当然可以求值，简短计算看出

$$\begin{aligned}\|e_n\|_A^2 &= e_n^{\mathrm{T}}Ae_n = (x_* - x_n)^{\mathrm{T}}A(x_* - x_n)\\ &= x_n^{\mathrm{T}}Ax_n - 2x_n^{\mathrm{T}}Ax_* + x_*^{\mathrm{T}}Ax_*\\ &= x_n^{\mathrm{T}}Ax_n - 2x_n^{\mathrm{T}}b + x_*^{\mathrm{T}}b = 2\varphi(x_n) + 常数\end{aligned}$$

[297] 于是，$\varphi(x)$与$\|e\|_A^2$除了因子2和（未知）常数$x_*^{\mathrm{T}}b$之外是一样的. 像$\|e\|_A^2$一样，$\varphi(x)$必须在$x=x_*$惟一地达到其极小值（即，$-x_*^{\mathrm{T}}b/2$）.

CG迭代可以解释为，求$x\in\mathbb{R}^m$的二次函数极小值的迭代过程. 在每一步上，要计算在一维空间$x_{n-1}+\langle p_{n-1}\rangle$内所有$x$上求$\varphi(x)$极小化的迭代$x_n=x_{n-1}+\alpha_n p_{n-1}$. [已经证实，公式$\alpha_n=(r_{n-1}^{\mathrm{T}}r_{n-1})/(p_{n-1}^{\mathrm{T}}Ap_{n-1})$保证，在所有步长$\alpha$中在此意义下$\alpha_n$是最优的.] 使得CG迭代变得极为出色的是选择的搜索方向p_{n-1}具有如下特殊性质：$\varphi(x)$在$x_{n-1}+\langle p_{n-1}\rangle$上的极小化实际上是在整个$\mathcal{K}_n$上的极小化.

解$Ax=b$的CG迭代和求特征值的兰乔斯迭代之间有密切的类似. 如在第27讲中所讨论的，A的特征值是关于瑞利商$r(x)=(x^{\mathrm{T}}Ax)/(x^{\mathrm{T}}x)$的$x\in\mathbb{R}^m$的平稳值. 如在习题36.1所指出的，与兰乔斯迭代第n步有关的特征值估计（里茨值）是，当同样函数$r(x)$的平稳值，如果x限于克雷洛夫子空间$\mathcal{K}_n$. 这是在前面2页中指出的完全并行的性质：$Ax=b$的解x_*是标量值函数$\varphi(x)$在$\mathbb{R}^m$中的极小点，而CG迭代x_n是同样函数的极小点，如果x限于$\mathcal{K}_n$.

38.6　CG和多项式逼近

最近4讲的一个主题是克雷洛夫子空间迭代与矩阵多项式之间的联系. 阿诺尔迪和兰乔斯迭代求解阿诺尔迪/兰乔斯逼近问题（34.3），以及GMRES迭代求解GMRES逼近问题（35.10）. 对于CG，合适的逼近问题包含了误差的A-范数.

CG逼近问题　求$p_n\in P_n$使得

$$\|p_n(A)e_0\|_A = 极小值. \tag{38.8}$$

其中e_0表示初始误差，$e_0=x_*-x_0=x_*$，P_n是次数$\leq n$的具有$p(0)=1$的多项式p的集合，它的定义如（35.7）. 从定理38.2，可以推导出下面的收敛定理.

定理38.3　*如果CG迭代在n步前没有收敛（即，$r_{n-1}\neq 0$），那么（38.8）有惟一解$p_n\in P_n$，并且对于同样的多项式p_n，迭代x_n有误差$e_n=p_n(A)e_0$. 因此有*

$$\frac{\|e_n\|_A}{\|e_0\|_A} = \inf_{p\in P_n}\frac{\|p(A)e_0\|_A}{\|e_0\|_A} \leq \inf_{p\in P_n}\max_{\lambda\in\Lambda(A)}|p(\lambda)|, \tag{38.9}$$

[298] *其中$\Lambda(A)$表示A的谱.*

证明　从定理38.1可以推得，对于某一$p\in P_n$有$e_n=p(A)e_0$. 在（38.9）中的等式是此式和定理38.2的结果. 至于在（38.9）中的不等式，如果$e_0=\sum_{j=1}^m a_jv_j$是e_0

在 A 的正交单位特征向量中的展开式，那么有 $p(A)e_0=\sum_{j=1}^{m}a_jp(\lambda_j)v_j$，于是

$$\|e_0\|_A^2=\sum_{j=1}^{m}a_j^2\lambda_j,\|p(A)e_0\|_A^2=\sum_{j=1}^{m}a_j^2\lambda_j(p(\lambda_j))^2.$$

这些恒等式推出了 $\|p(A)e_0\|_A^2/\|e_0\|_A^2\leqslant\max_{\lambda\in\Lambda(A)}|p(\lambda)|^2$，此式隐含问题中的不等式. □

38.7 收敛速度

定理 38.3 建立了 CG 迭代的收敛速度是由 A 的谱的位置所决定的这一结论. 一个好的谱是在其上多项式 $p_n\in P_n$ 可以很小，并且其大小随 n 快速减小. 粗略地说，这对于两个原因中的一个或两个都可能发生：特征值可以聚集在小的簇内，或者它们位于在相对意义下与原点很好分离的位置. 定理 38.3 的两个最著名的推论在其极端形式下提出了这两个思想.

首先，假定特征值完全聚集在一起，但对于聚集于何处不作假定.

定理 38.4　*如果 A 仅有 n 个不同特征值，那么 CG 迭代至多在 n 步内收敛.*

证明　由于存在多项式 $p(x)=\prod_{j=1}^{n}(1-x/\lambda_j)\in P_n$，该多项式在任意给定的几个点 $\{\lambda_j\}$的集合上为零，所以这是（38.9）的推论. □

在另一极端情形，假定不知道特征值的任何聚类，但仅知道它们离原点的距离至多以因子 $\mathcal{K}\geqslant 1$ 变化. 换言之，假设仅知道 2-范数条件数 $\mathcal{K}=\lambda_{\max}/\lambda_{\min}$，其中 $\lambda_{\max}$ 和 $\lambda_{\min}$是 A 的极特征值.

定理 38.5　*设把 CG 迭代应用到对称正定矩阵问题 $Ax=b$，其中 A 有 2-范数条件数 $\mathcal{K}$. 那么误差的 A-范数满足*

$$\frac{\|e_n\|_A}{\|e_0\|_A}\leqslant 2\bigg/\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^n+\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^{-n}\right]\leqslant 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n. \tag{38.10}$$

299

证明　利用定理 38.3，足以求得一个多项式 $p\in P_n$，对于 $\lambda\in[\lambda_{\min},\lambda_{\max}]$其最大值是（38.10）中间表达式. 选择的多项式是有比例的且有位移的切比雪夫多项式 $p(x)=T_n(\gamma-2x/(\lambda_{\max}-\lambda_{\min}))/T_n(\gamma)$，其中 T_n 是通常的 n 次切比雪夫多项式，γ 取为特殊的值 $\gamma=(\lambda_{\max}+\lambda_{\min})/(\lambda_{\max}-\lambda_{\min})=(\kappa+1)/(\kappa-1)$. 对于 $x\in[\lambda_{\min},\lambda_{\max}]$，$p(x)$分子中 T_n 的自变量位于［-1，1］内，这表示这分子的量值≤1. 因此，为了证明这个定理，指出下式就够了.

$$T_n(\gamma)=T_n\left(\frac{\kappa+1}{\kappa-1}\right)=\frac{1}{2}\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^n+\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^{-n}\right]. \tag{38.11}$$

通过作变量变换 $x=\frac{1}{2}(z+z^{-1}),T_n(x)=\frac{1}{2}(z^n+z^{-n})$，在切比雪夫多项式研究中是标

准的，就可以做到这一点. 如果$(\kappa+1)/(\kappa-1)=\frac{1}{2}(z+z^{-1})$，即，$\frac{1}{2}z^2-(\kappa+1)/(\kappa-1)z+\frac{1}{2}=0$，那么得到一个二次方程，其根为

$$z=\left(\frac{\kappa+1}{\kappa-1}\right)+\sqrt{\left(\frac{\kappa+1}{\kappa-1}\right)^2-1}=\frac{\kappa+1+\sqrt{(\kappa+1)^2-(\kappa-1)^2}}{\kappa-1}$$

$$=\frac{\kappa+1+\sqrt{4\kappa}}{\kappa-1}=\frac{(\sqrt{\kappa}+1)^2}{(\sqrt{\kappa}+1)(\sqrt{\kappa}-1)}=\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}.$$

于是，正如所要求的，对于这个z值，$T_n(\gamma)=\frac{1}{2}(z^n+z^{-n})$，这就是（38.11）. □

定理38.5是CG迭代收敛性最著名的结果. 由于
当$\kappa\to\infty$时

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\sim 1-\frac{2}{\sqrt{\kappa}}$$

这就意味着如果κ是大的. 但不太大，则收敛到确定的容许限可以期望在$O(\sqrt{\kappa})$次迭代达到. 但必须记住，这仅是一个上界. 对于特殊的右手边向量（不那么普通的）或若谱是聚类的（更普通的），收敛可以更快.

38.8　例　　子

说到CG收敛的例子，考虑结构如下的500×500稀疏矩阵A. 首先在每一对角线位置上置1. 在非对角线位置用$[-1,1]$上均匀分布的随机数（保持对称性$A=A^{\mathrm{T}}$）置数. 然后，当非对角线元素满足$|a_{ij}|>\tau$时以零代替，其中τ是一个参数. 由于τ接近零，结果是良态的正定矩阵，其非零元素的密度近似地为τ. 当τ增加时，条件数和稀疏性两者都变坏.

300

图38-1画出了对应于具有$\tau=0.01, 0.05, 0.1, 0.2$的这个类型矩阵的CG迭代20步的收敛曲线.（右手边$\boldsymbol{b}$取为随机向量.）对于$\tau=0.01$，A有3092个非零元素并且条件数$\kappa\approx1.06$. 在第9步收敛到机器精度，大约6×10^4次flop. 对于$\tau=0.05$，A有13062个非零元素且条件数$\tau\approx1.83$，需要用19步收敛，大约5×10^5次flop，对于$\tau=0.1$，有25526个非零元素和条件数$\tau\approx10.3$，20步后仅5位数字收敛和10^6次flop. 对于$\tau=0.2$，有50834个非零元素，完全不收敛. 现在最小的特征值是负的，所以A不再是正定的并且不适合用CG迭代.（事实上，CG迭代经常成功地用于不定矩阵，但在此情况下，矩阵不仅是不定的而且还是病态的.）

注意，图38-1的$\tau=0.01$曲线如此精密地与图32-1中图解的理想描绘相一致！对于这个例子，6×10^4次flop的运算计数以大约700的倍数胜过楚列斯基因子分解

(23.4). 不幸的是，不是每个在实际中出现的矩阵有如此良态的谱，甚至在尽了最大努力求出了一个好的预处理器以后仍是如此. 301

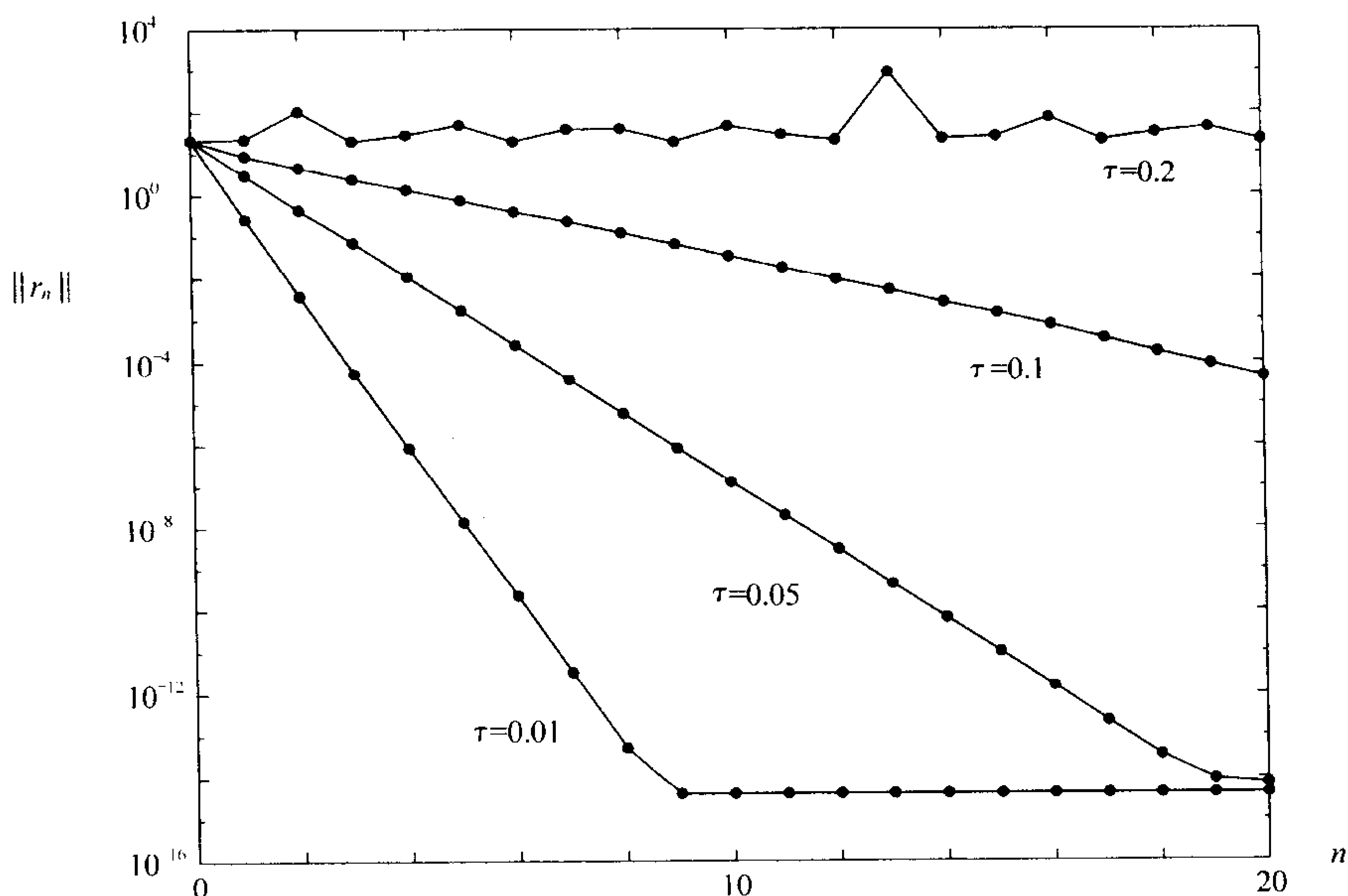


图 38-1　对于在正文中描述的 500×500 稀疏矩阵 $\boldsymbol{A}$ 的 CG 收敛曲线. 对于 $\tau=0.01$，用 CG 解这个方程组大约比用楚列斯基因子分解来解快 700 倍. 对于 $\tau=0.2$，矩阵不是正定的并且 CG 方法不收敛

习　题

38.1　基于在正文中给出的条件数 κ，确定图 38-1 以 $\tau=0.01, 0.05, 0.1$ 的矩阵 $\boldsymbol{A}$ 用定理 38.5 所预期的收敛速度. 在图 38-1 的复制图上画出表示这些预期的收敛速度如何精密地与真实的收敛速度相一致的直线.

38.2　设 $\boldsymbol{A}$ 为实对称的 805×805 矩阵，其特征值为 $1.00, 1.01, 1.02, \cdots, 8.98, 8.99, 9.00$ 以及 $10, 12, 16, 24$. 必须取共轭梯度迭代的多少步才能保证以 10^6 因子缩减初始误差 $\|\boldsymbol{e}_0\|_A$?

38.3　把共轭梯度应用到对称正定矩阵 $\boldsymbol{A}$，并有结果 $\|\boldsymbol{e}_0\|_A=1$, $\|\boldsymbol{e}_{10}\|_A=2\times2^{-10}$. 仅基于这些数据.

(a) 在 $\kappa(\boldsymbol{A})$ 上能够给出什么样的界值?

(b) 在 $\|\boldsymbol{e}_{20}\|_A$ 上能够给出什么样的界值?

38.4　假设 $\boldsymbol{A}$ 是稠密的对称正定 1000×1000 矩阵，$\kappa(\boldsymbol{A})=100$. 粗略地估计以 10 位数字精度用 (a) 楚列斯基因子分解，(b) 最优参数 α 的理查森迭代（习题 35.3）和(c) CG解 $\boldsymbol{Ax}=\boldsymbol{b}$ 需要多少次 flop.

38.5　已经把 CG 描述为（38.7）的函数 $\varphi(x)$ 的迭代极小化. 极小化同一函数的另外一个方法（一般非常慢）是最速下降（Steepest descent）方法.

(a) 导出 $\varphi(x)$ 的梯度的公式 $\nabla\varphi(x) = -r$，于是，最速下降迭代法相应于选择 $p_n = r_n$ 来代替算法 38.1 中 $p_n = r_n + \beta_n p_{n-1}$.

(b) 确定最速下降迭代最优步长 α_n 的公式.

(c) 写出完全的最速下降迭代，在主循环内有 3 个运算.

38.6　设 $\boldsymbol{A}$ 为 100×100 的三对角对称矩阵，其对角线上的元素为 $1,2,\cdots,100$，在上-下对角线上的元素为 1 并设 $\boldsymbol{b}=(1,1,\cdots,1)^{\mathrm{T}}$. 写出一个近似求解 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的 CG 和最速下降迭代 100 步的程序. 画出具有 4 条曲线的一个图形：关于 CG 的计算的剩余范数 $\|r_n\|_2$，关于 CG 的实际的剩余范数 $\|\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_n\|_2$，最速下降的剩余范数 $\|r_n\|_2$ 以及定理 38.5 的估计 $2(\sqrt{\kappa}-1)^n/(\sqrt{\kappa}+1)^n$. 对上述结果作出评论.

302

第 39 讲　双正交化方法

不是所有的非对称方程组的克雷洛夫子空间迭代都存在时间的增加和成本消耗的增长. 已经设计出了基于三项递推的方法, 并且它们是目前可采用的最强有力的非对称迭代, 至少对于这一类的一些迭代, 所付出的代价是必须工作于由 A^* 的乘法以及 A 的乘法产生的两个克雷洛夫子空间而非一个子空间.

39.1　内容定位

在 32.3 节给出了克雷洛夫子空间矩阵迭代的表:

	$Ax=b$	$Ax=\lambda x$
$A=A^*$	CG	兰乔斯
$A\neq A^*$	GMRES CGN BCG 等	阿诺尔迪

这些表中的其中 3 个单元格的讨论已完成, 至于第 4 个, 左下位置, 已经讨论了 GMRES. 因此在本讲中, 转向该单元格的最后二行, 在这个简单的且容易分析的算法 CGN 上又花费少量时间, 然后转向主要的题目, "BCG et al." ——双正交化方法.

39.2　应用到正规方程的 CGN = CG

设 $A\in\mathbb{C}^{m\times m}$ 非奇异但不必定是埃米尔特的, 所以对于任何 $b\in\mathbb{C}^m$, $Ax=b$ 是一个非奇异的方形方程组. 解这样的方程组的最简单方法之一是把 CG 迭代应用到正规方程 (11.9),

$$A^*Ax = A^*b. \tag{39.1}$$

(矩阵 A^*A 不是显式地形成的, 它需要 m^3 次 flop, 事实上, 每个矩阵—向量乘积 A^*Av是用如 $A^*(Av)$这样的两步来计算的.) 由于 A 非奇异, 所以 A^*A 是埃米尔特正定的, 或如果 A 是实的, 那么 A^*A 是对称正定的. 于是上一讲的定理可以应用, 并且如果 A^*A 的特征值合适地分布, 那么可期望快速收敛, 这个方法以 CGN (也是 CGNR) 的命名, 这粗略地代表"应用到正规方程的 CG".

由于已经分析了 CG 的性质, 所以理解 CGN 的性质不需要新的东西, 如果如在算法 38.1 中一样, 初始猜测是 $x_0=0$, 那么从定理 38.1 可以看到, 后者的迭代属于

由 A^*A 产生的克雷洛夫子空间：

$$x_n \in \langle A^*b, (A^*A)A^*b, \cdots, (A^*A)^{n-1}A^*b \rangle \tag{39.2}$$

从定理38.2可知，在每一步，误差的 A^*A-范数是在这个空间上的极小化，并且由于 $\|e_n\|_{A^*A}^2 = e_n^* A^* A e_n = \|Ae_n\|_2^2 = \|r_n\|_2^2$，这是表明剩余 $r_n = b - Ax_n$ 的2-范数是极小化：

$$\|r_n\|_2 = \text{极小值} \tag{39.3}$$

的另一种方法，于是像GMRES一样，CGN是最小剩余方法，但因为克雷洛夫子空间（33.5）和（39.2）是不同的，所以这两种方法决不是等价的.

根据定理38.3，CGN的收敛性是由 A^*A 的特征值来控制的，这些数等于 A 的奇异值的平方. 于是，CGN的收敛性是由 A 的奇异值来确定的，并且原则上与 A 的特征值毫不相干. 然而，包含了平方这一事实是令人遗憾的. 如果 A 有条件数 κ，那么 A^*A 有条件数 κ^2，并且对于CGN，（38.10）的模拟变成

$$\frac{\|r_n\|_2}{\|r_0\|_2} \leqslant 2\left(\frac{\kappa-1}{\kappa+1}\right)^n. \tag{39.4}$$

对于大的 κ，这比（38.10）更糟糕，因为这意味着对于收敛到固定的精度需要 $O(\kappa)$ 而不是 $O(\sqrt{\kappa})$ 次迭代. 304

这个“条件数的平方”已经给了CGN迭代一个不好的名声，总的说来，这是理所应得的. 尽管如此，对于某些问题，CGN大大地胜过其他可能的方法，其原因是它们的收敛性依赖于特征值而不是奇异值，所需的全部是其奇异值是良态的而其特征值不是良态的矩阵，例如一个良态矩阵，其谱在复平面内围绕原点分布. 一个极端的例子由下面形式的 $m \times m$ 循环矩阵

$$A = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & 0 & 1 & \\ & & & 0 & 1 \\ 1 & & & & 0 \end{bmatrix}. \tag{39.5}$$

所给出. 这个矩阵的奇异值都等于1，但其特征值是单位1的 m 次方根. 对于一般的右手边 b，对于收敛，GMRES需要 m 步而CGN一步就可收敛.（见习题39.1）

CGN迭代的另一个优点是，由于它基于正规方程，所以它可不作修改地应用到最小二乘方问题（参见算法11.1），其中 A 不再是方阵. 另一方面，最小二乘问题的一些迭代方法是基于习题19.1的块方程组（19.4）的.

39.3　三对角的双正交化

如在第36讲中已看到的那样，兰乔斯迭代是三对角的正交化过程. 如果执行了完全

的 m 步（在精确运算中），那么将产生埃米尔特矩阵到三对角线型的酉约化：$A=QTQ^*$.

如果 A 不是埃米尔特的，那么这样的约化一般是不可能的：不是必须放弃酉变换就是必须放弃最后的三对角线型. 阿诺尔迪迭代，海森伯格正交化过程，就适合于后者. 如果执行了完全的 m 步，那么将产生任意方矩阵到海森伯格型的酉约化 (33.12)：$A=QHQ^*$.

双正交化方法是基于相反的选择. 若坚持三对角线的结果而放弃用酉变换，那么有三对角线的双正交化过程：$A=VTV^{-1}$，其中 V 是非奇异的，但一般不是酉的（图 39-1）. 取其伴随给出等价方程 $A^*=V^{-*}T^*(V^{-*})^{-1}$. （从原书 12 页回想起 $V^{-*}=(V^*)^{-1}=(V^{-1})^*$.）术语"双正交"参考如下事实，虽然 V 的列并不相互正交，但它们与 V^{-*} 的列是正交的. 这简单地可由恒等式 $(V^{-*})^*V=V^{-1}V=I$ 来得到.

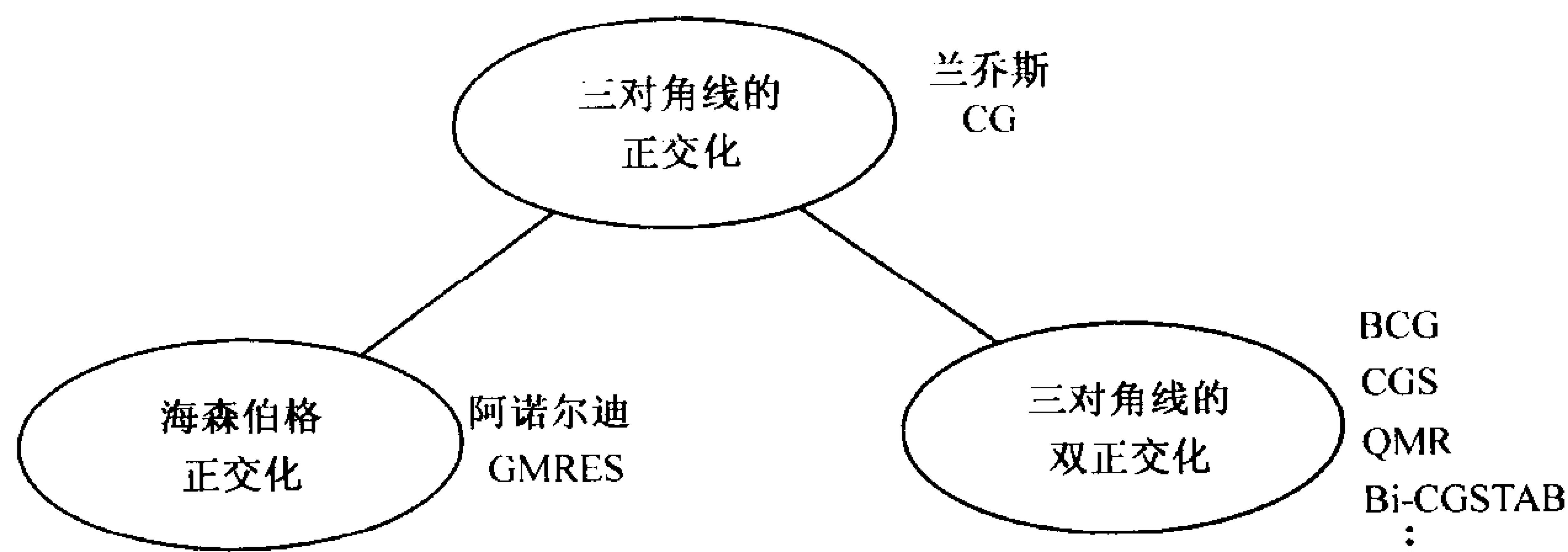


图 39-1　克雷洛夫子空间迭代分类. 如果矩阵是埃米尔特的（顶行），那么它可以用三项递推来正交化——三对角矩阵. 如果矩阵不是埃米尔特的，那么必须不是放弃三对角线结构就是放弃正交性

为了开始把这一思想变成迭代算法，必须了解对于 $n<m$ 所包含的内容. 设 V 是一个非奇异矩阵，满足 $A=VTV^{-1}$，其中 T 是三对角线的，并定义 $W=V^{-*}$. 令 v_j 和 w_j 分别表示 V 和 W 的第 j 列的列向量. 这些向量是在此意义 [305]

$$w_i^* v_j=\delta_{ij}, \tag{39.6}$$

下是双正交的，其中 δ_{ij} 是克罗内克 δ 函数. 对于每个 n，$1\leqslant n\leqslant m$，由 (33.1) 定义 $m\times n$ 矩阵

$$V_n=\left[\begin{array}{c|c|c} v_1 & \cdots & v_n \end{array}\right],\quad W_n=\left[\begin{array}{c|c|c} w_1 & \cdots & w_n \end{array}\right]. \tag{39.7}$$

以矩阵形式，双正交性条件可以写成

$$W_n^* V_n=V_n^* W_n=I_n,$$

其中 $\boldsymbol{I}_n$ 是 n 维的单位矩阵.

现在能够写出作为双正交化方法基础的关键公式. 对于兰乔斯迭代，有（36.5）和（36.6），

$$\boldsymbol{A}\boldsymbol{Q}_n = \boldsymbol{Q}_{n+1}\tilde{\boldsymbol{T}}_n, \qquad \boldsymbol{T}_n = \boldsymbol{Q}_n^*\boldsymbol{A}\boldsymbol{Q}_n.$$

对于阿诺尔迪迭代，有（33.12）和（33.13），

306

$$\boldsymbol{A}\boldsymbol{Q}_n = \boldsymbol{Q}_{n+1}\tilde{\boldsymbol{H}}_n, \qquad \boldsymbol{H}_n = \boldsymbol{Q}_n^*\boldsymbol{A}\boldsymbol{Q}_n.$$

对于双正交化方法有对应的公式：

$$\boldsymbol{A}\boldsymbol{V}_n = \boldsymbol{V}_{n+1}\tilde{\boldsymbol{T}}_n, \tag{39.8}$$

$$\boldsymbol{A}^*\boldsymbol{W}_n = \boldsymbol{W}_{n+1}\tilde{\boldsymbol{S}}_n, \tag{39.9}$$

$$\boldsymbol{T}_n = \boldsymbol{S}_n^* = \boldsymbol{W}_n^*\boldsymbol{A}\boldsymbol{V}_n. \tag{39.10}$$

其中 $\boldsymbol{V}_n$ 和 $\boldsymbol{W}_n$ 有维数 $m\times n$，$\tilde{\boldsymbol{T}}_{n+1}$和$\tilde{\boldsymbol{S}}_{n+1}$是维数为$(n+1)\times n$的（非埃米尔特）三对角矩阵，以及 $\boldsymbol{T}_n=\boldsymbol{S}_n^*$ 是由删去$\tilde{\boldsymbol{T}}_{n+1}$的最后一行或删去$\tilde{\boldsymbol{S}}_{n+1}^*$的最后一列得到的 $n\times n$ 矩阵.

类似于原书252页的叙述，(39.8) 可以呈现如下，

$$\left[\quad \boldsymbol{A} \quad\right]\left[\boldsymbol{v}_1\,\middle|\,\cdots\,\middle|\,\boldsymbol{v}_n\right] = \left[\boldsymbol{v}_1\,\middle|\,\cdots\,\middle|\,\boldsymbol{v}_{n+1}\right]\begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \gamma_{n-1} \\ & & & \beta_{n-1} & \alpha_n \\ & & & & \beta_n \end{bmatrix},$$

此对应于三项递推关系

$$\boldsymbol{A}\boldsymbol{v}_n = \gamma_{n-1}\boldsymbol{v}_{n-1} + \alpha_n\boldsymbol{v}_n + \beta_n\boldsymbol{v}_{n+1}. \tag{39.11}$$

类似地，(39.9) 取成形式

$$\left[\quad \boldsymbol{A}^* \quad\right]\left[\boldsymbol{w}_1\,\middle|\,\cdots\,\middle|\,\boldsymbol{w}_n\right] = \left[\boldsymbol{w}_1\,\middle|\,\cdots\,\middle|\,\boldsymbol{w}_{n+1}\right] = \begin{bmatrix} \bar{\alpha}_1 & \bar{\beta}_1 & & & \\ \bar{\gamma}_1 & \bar{\alpha}_2 & \bar{\beta}_2 & & \\ & \bar{\gamma}_2 & \bar{\alpha}_3 & \ddots & \\ & & \ddots & \ddots & \bar{\beta}_{n-1} \\ & & & \bar{\gamma}_{n-1} & \bar{\alpha}_n \\ & & & & \bar{\gamma}_n \end{bmatrix},$$

对应地，

$$A^* w_n = \bar{\beta}_{n-1} w_{n-1} + \bar{\alpha}_n w_n + \bar{\gamma}_n w_{n+1}. \tag{39.12}$$

（由于在前面 3 讲中，假定 A 是实的，所以对于复共轭，以前没有出现过横线.）

如克雷洛夫子空间迭代一样，这些方程提示了一个算法. 开始任取向量 v_1 和 w_1 但使其满足 $v_1^* w_1 = 1$，并令 $\beta_0 = \gamma_0 = 0, v_0 = w_0 = \mathbf{0}$. 现在，对于 $n = 1, 2, \cdots$，令 $\alpha_n = w_n^* A v_n$，由（39.6）和（39.11）或（39.12）一样可以推出. 然后，由（39.11）和（39.12）可以确定向量 v_{n+1} 和 w_{n+1} 到标量因子. 这些因子可以任意选择，只要满足正规化条件 $w_{n+1}^* v_{n+1} = 1$ 即可，然而 β_{n+1} 和 γ_{n+1} 则由（39.11）和（39.12）来确定. 307

由刚描述的过程产生的向量存在于克雷洛夫子空间之中：

$$v_n \in \langle v_1, Av_1, \cdots, A^{n-1} v_1 \rangle, w_n \in \langle w_1, A^* w_1, \cdots, (A^*)^{n-1} w_1 \rangle. \tag{39.13}$$

对于一般的矩阵，在精确运算中，这个过程将在 m 步后运行完成，但对于某些特殊矩阵，在达到此点之前，这个过程可能失败. 如果在某一步有 $v_n = \mathbf{0}$ 或 $w_n = \mathbf{0}$，那么 A 或 A^* 的不变子空间已经找到：三对角矩阵 T 是可约的（参考习题 33.2）. 另一方面，除了 $w_n^* v_n = \mathbf{0}$. 以外，$v_n \neq \mathbf{0}$ 和 $w_n \neq \mathbf{0}$ 也可以发生. 这种更严重类型的计算失败的可能性会出现在大多数的双正交化方法中，而在数值分析的其他领域，对某些问题可能完全失败的事实意味着，许多在浮点运算中可能有潜在的不利结果的问题可以发生近似失败. 具有这些现象的某些方法将在本讲末讨论.

39.4 BCG = 双共轭梯度

利用刚描述的双正交化过程的一个途径就是计算特征值：当 $n \to \infty$ 时，T_n 的某些特征值可以很快地收敛到 A 的特征值. 另外一个应用是（现仅简单地讨论）解非奇异的方程组 $Ax = b$. 这种类型的典型算法称为**双共轭梯度**（biconjugate gradients）或 BCG.

BCG 的原理如下：取 $v_1 = b$，所以（39.13）中的第一个克雷洛夫子空间变成 $\mathcal{K}_n = \langle b, Ab, \cdots, A^{n-1} b \rangle$. 回想 GMRES 的原理是，取 $x_n \in \mathcal{K}_n$ 使得正交化条件满足

$$\text{GMRES}: r_n \perp \langle Ab, A^2 b, \cdots, A^n b \rangle = A\,\mathcal{K}_n \tag{39.14}$$

其中 $r_n = b - Ax_n$ 是对应于 x_n 的剩余（图 35-1）. 这个选择有极小化 $\|r_n\|$，剩余的 2-范数的效果. BCG 算法的原理是在同样的子空间中取 x_n，$x_n \in \mathcal{K}_n$，但实施正交化条件

$$\text{BCG}: r_n \perp \langle w_1, A^* w_1, \cdots, (A^*)^{n-1} w_1 \rangle \tag{39.15}$$

其中 $w_1 \in \mathbb{C}^m$ 是满足 $w_1^* v_1 = 1$ 的任意向量. 在应用中，有时取 $w_1 = v_1 / \|v_1\|_2$. 与

(39.14) 不同，这个选择并没有极小化$\|r_n\|_2$，并且从极小化迭代次数的观点来看，308它不是最优的. 其好处是它可以用三项递推而不是用 GMRES 的 $(n+1)$项递推来实施.

将不给出详细的推导，现以其标准形式来表示 BCG 算法. 请将下面的算法与算法 38.1，即 CG 算法加以比较. 这两个算法除了 CG 的搜索方向序列$\{p_n\}$变成两个序列$\{p_n\}$和$\{q_n\}$，以及 CG 的剩余序列$\{r_n\}$变为两个序列$\{r_n\}$和$\{s_n\}$之外，其余是一样的.

算法 39.1　双共轭梯度 (BCG) 迭代

$x_0=0, p_0=r_0=b, q_0=s_0=$任给

for $n=1,2,3,\cdots$

$$\alpha_n=(s_{n-1}^*r_{n-1})/(q_{n-1}^*Ap_{n-1})$$

$$x_n=x_{n-1}+\alpha_n p_{n-1}$$

$$r_n=r_{n-1}-\alpha_n Ap_{n-1}$$

$$s_n=s_{n-1}-\bar{\alpha}_n A^* q_{n-1}$$

$$\beta_n=(s_n^* r_n)/(s_{n-1}^*r_{n-1})$$

$$p_n=r_n+\beta_n p_{n-1}$$

$$q_n=s_n+\bar{\beta}_n q_{n-1}$$

如在定理 38.1 中的，已经指出对于$j<n$，有$s_n^* r_j=0$和$q_n^* Ap_j=0$.

39.5　例　　子

在图 38-1 中说明了依赖于参数τ的 500×500 稀疏对称正定矩阵的 CG 迭代的收敛性. 为了说明 BCG 的收敛性，考虑同样矩阵但作一个改变：所有元素的符号是随机化的. 这使得矩阵不再是埃米尔特的，并对角线上的优势元素随机地为 1 和 -1，而不是都是 1，所以其特征值是围绕 1 和 -1 聚集而不是刚好为 1.

图 39-2 指出了对于$\tau=0.01$的这样的矩阵的 GMRES 和 BCG 的收敛性. 首先考虑 GMRES 曲线，注意到其收敛如图 38-1 所指出的一半那么快，并在每一奇数步上基本无进展，但在每一偶数步上有均匀的进展. 这个奇-偶效应是矩阵的近似±对称性的结果：具有$p(0)=1$的$2k+1$次多项式$p(z)$在 1 和 -1 处可以不小于相应的$2k$次多项式. 现在转向 BCG 曲线，可以看到在整体意义下收敛是可以比较的，但它不再是单调的了，在每一奇数步上，显示了在数量上大到大约10^2的峰值. 最后得到的精度也蒙受了 1 位数以上的损失. 所有这些特性是 BCG 计算的典309型特性.

在图 39-2 中水平轴是步数 n，这与计算的工作量不一样．在每一步，GMRES 需要包含 A 的矩阵-向量乘法，而 BCG 需要包含 A 和 A^* 的乘法．对于矩阵-向量乘法占主要工作且有足够的存储可利用的问题，GMRES 必然要比 BCG 快两倍或更快．然而，这里的矩阵足够稀疏以致于与处理长的递推有关的工作量占有重要地位，事实上，图 39-2 的 BCG 计算速度是以优于 2 的因子快于 GMRES 计算．

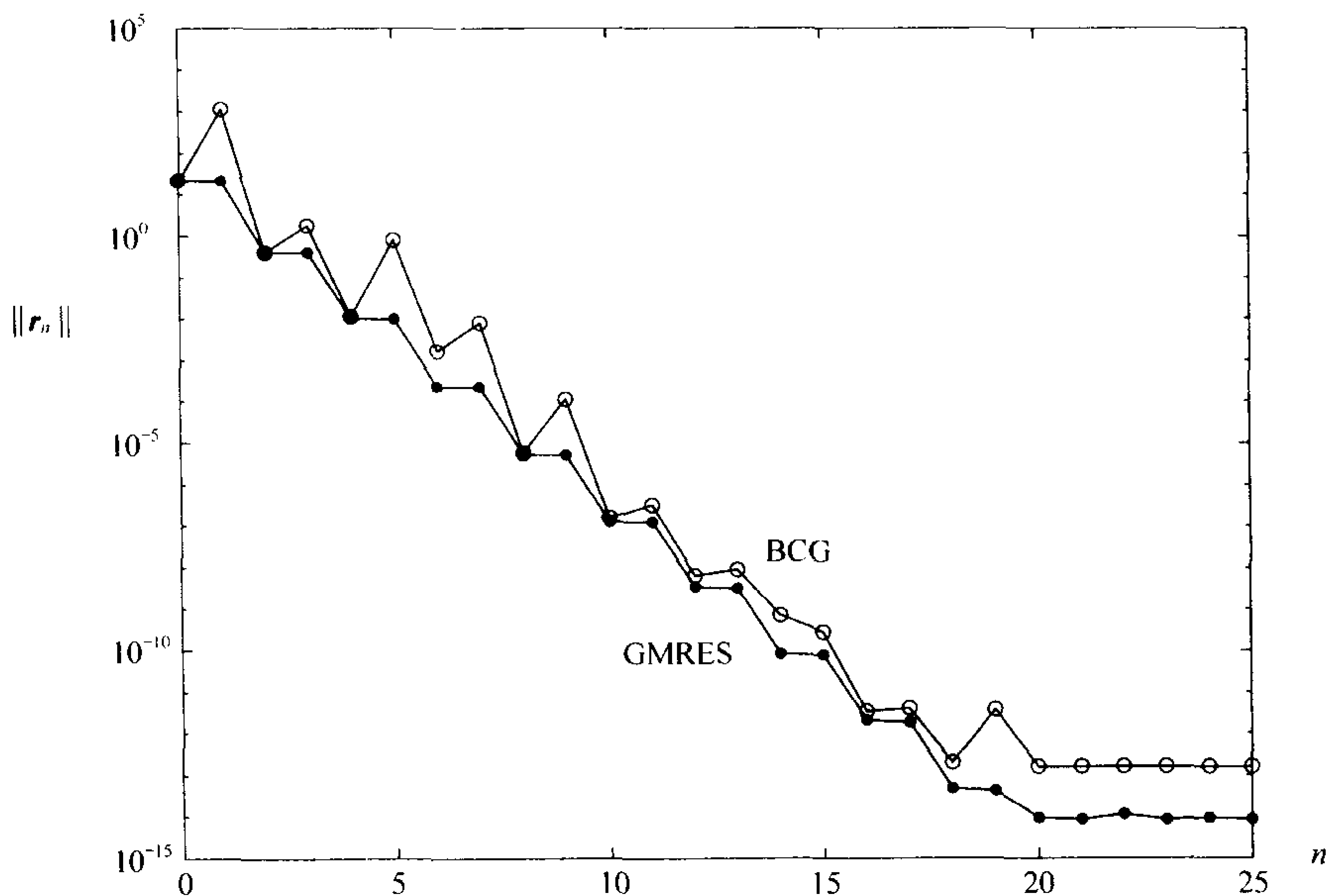


图 39-2 GMRES 和 BCG 的比较，其矩阵为图 38-1 中标定 $\tau=0.01$ 的 500×500 矩阵，但其元素的符号是随机的

39.6 QMR 和其他变式

BCG 比起 GMRES 来有一个显著的优点：它包含三项递推计算，能够使每步的工作和存储保持在一种可控状态，甚至当需要许多步时亦如此．另一方面，它有两个缺点，其一是作为步数的函数，与 GMRES 的单调性且常常快速地收敛相比较，它的收敛较慢，并且常常是没有规律的，有时比图 39-2 中显示的还没有规律．不规则的收敛是没有吸引力的，并且由于舍入误差的原因，它会最终使可能达到的精度下降（习题 39.4）．在极端的情况下，即使方程组是良态的，也可能在迭代中内积变为零，因此不能进一步进行，由此它成为了破坏迭代的现象． 310

BCG 的另一个问题是它需要用 A^* 和 A 来作乘．考虑到这些乘积如何在数学上以及利用计算机结构来实现，这会成为或大或小的一个负担，甚至于完全不可能做到．

为解决这两个问题，从 20 世纪 80 年代开始，人们就提出了许多 BCG 的变形，

下面是其中最著名的几个. 参考文献在评注中给出.

向前看（Look-ahead）兰乔斯法（Parlett，Taylor 和 Liu，1985）.

CGS = 平方的共轭梯度法（Sonneveld,1989）.

QMR = 拟-极小剩余法（Freund 和 Nachtigal,1991）.

Bi-CGSTAB = 稳定的 BCG 法（Van der Vorst,1992）.

TFQMR = 无转置 QMR 法（Freund,1993）.

下面对这些方法仅稍作说明，并不详细论述.

向前看兰乔斯算法基于如下事实，当失败就要发生时，立刻用两步或多步迭代取代单步迭代来避免失败. Parlett 等人的思想后来由其他人进行了多方面的发展，并且并入了后人推荐的 QMR 算法中. 能证明失败的现象等价于在函数的 Padé 逼近表中恒等元素的平方块的现象，并且向前看思想相当于在一步中穿过这种块的方法. 当然，在使用中，对于完全的失败是不好试验的，可以用适当的容许来定义近似失败.

CGS 算法是基于下面的发现，如果 BCG 的两步以不同的方式组成一步，以致算法是“平方的”，那么可避免 A^* 的乘法. 这个结果是“无转置”方法，此方法尽管收敛有加倍的不规则，但有时其收敛比 BCG 快 1 倍.

QMR 算法基于如下观察，虽然三项递推不能用作极小化 $\|r_n\|$，但它们可以用作极小化一个不同的、依赖数据的范数，此范数在实际中通常离 $\|r_n\|$ 不很远. 这大大地减少了舍入误差的影响，对于收敛的光滑性有显著的效果. Bi-CGSTAB 算法是另一个也大大地光滑 BCG 的收敛速率的方法，TFQMR 是 QMR 的变式，它把光滑收敛性与避免 A^* 的需要结合起来.

最近，研究工作是针对光滑的收敛曲线、为避免失败的向前看以及无转置运算这 3 个优点进行组合. 至今，这三者还没有完全满意地组合成一个单一的算法，但这个研究领域是新的. [311]

习　题

39.1　考虑 m 维矩阵（39.5）的问题 $Ax = b$.

(a) 证明奇异值都为 1，并证明这暗含着 CGN 一步就收敛.

(b) 证明特征值为单位的 m 次方根，并且证明这意味着对于一般的 b，GMRES 需要 m 步收敛.

(c) 这个矩阵 A 具有如此的结构以致于理解其收敛性质不用考虑特征值或奇异值. 特别地，用初等理由说明为什么对于右手边 $b = (1,0,0,\cdots,0)^T$ GMRES 要 m 步收敛.

39.2　作为习题 39.1 的反例. 设计出任意 m 维具有几乎相反性质的矩阵的一个例子：GMRES 在两步内收敛，但 CGN 需要 m 步.

39.3　(a) 如果 A 是埃米尔特的，适当地选取 s_0，那么算法 39.1 约化到算法 38.1. 证明这一

陈述并确定这个适当的 s_0.

(b) 假设 $\boldsymbol{A}$ 是一个对称的但不是埃米尔特的复矩阵，证明，对 $\boldsymbol{s}_0$ 的一个不同选取，算法 39.1 再次约化到仅包含一个三项递推的迭代.

39.4 图 39-2 说明了，如果双正交化方法的收敛曲线中有峰值，那么这可以影响到浮点运算中可达到的精度. 不作严格的论证，说明为什么那样，并且说明与高斯消元（第 22 讲）中的增长因子有类似之处.

39.5 对于下述 $\boldsymbol{m}\times\boldsymbol{m}$ 问题 $\boldsymbol{Ax}=\boldsymbol{b}$，你期望 CG，GMRES，CGN 或 BCG 中的哪一个最有效，为什么?

(a) $\boldsymbol{A}$ 为 $m=10^4$ 的稠密的非埃米尔特矩阵，其特征值除 3 个之外均近似等于 -1.

(b) 同上，但其特征值除了 3 个之外均散布于区域 $-10\leqslant\mathrm{Real}(\lambda)\leqslant 10$，$-1\leqslant\mathrm{Imag}(\lambda)\leqslant 1$.

(c) $\boldsymbol{A}$ 为 $m=10^6$ 的，且仅有 10^7 个非零元素的稀疏的非埃米尔特矩阵，其特征值如 (a).

(d) $\boldsymbol{A}$ 为 $m=10^5$ 的稀疏的埃米尔特矩阵，其特征值散布在区间 [1，100] 上.

(e) 除了离群特征值 0.01 和 10000 外同上题.

(f) 除了外加的离群特征值 -1，-10 和 -100 外同上题.

(g) $\boldsymbol{A}$ 为 $m=10^5$ 的稀疏、正则矩阵，其特征值是散布在圆环 $1\leqslant|\lambda|\leqslant 2$ 上的复数.　312

第40讲 预 处 理

矩阵迭代的收敛性依赖于矩阵的性质——特征值、奇异值有时还需要其他的信息. 人们发现，在许多情况下，可以对感兴趣的问题加以转换使得矩阵的性质获得巨大的改进. 这个发现实际上是使得20世纪70年代和80年代许多方法不断涌现的原因之一. “预处理”的这个过程对于大多数迭代方法的成功应用是必不可少的.

40.1 $\boldsymbol{Ax}=\boldsymbol{b}$ 的预处理器

理论上，对方程组预处理的思想是初等的. 设要求解 $m\times m$ 非奇异方程组

$$\boldsymbol{Ax} = \boldsymbol{b}. \tag{40.1}$$

对于任意的非奇异 $m\times m$ 矩阵 $\boldsymbol{M}$，方程组

$$\boldsymbol{M}^{-1}\boldsymbol{Ax} = \boldsymbol{M}^{-1}\boldsymbol{b} \tag{40.2}$$

有相同的解. 然而，如果迭代求解（40.2)，那么收敛将依赖于 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 的性质而不是 $\boldsymbol{A}$ 的性质. 如果这个预处理器（preconditioner）选择得很好，那么求解（40.2）可以比求解（40.1）更快.

313 当然，为了使这个思想有用，必须使有效地计算由乘积 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 表示的运算成为可能. 如在数值线性代数中一样，这决不意味着逆矩阵 $\boldsymbol{M}^{-1}$ 是显式构造的，而是求解下面形式的方程组

$$\boldsymbol{My} = \boldsymbol{c}. \tag{40.3}$$

很快想起两个极端的情形. 如果 $\boldsymbol{M}=\boldsymbol{A}$，那么（40.3）与（40.1）一样，所以应用这个预处理器与求解原来的问题一样困难，并且没有得到什么收获. 如果 $\boldsymbol{M}=\boldsymbol{I}$，那么（40.2）与（40.1）一样，所以应用这预处理器是不重要的，它也未做什么事情. 在这两个极端情形之间存在有用的预处理器，这个预处理器足够结构化以致很快能求解（40.3)，但它在某些意义下充分靠近 $\boldsymbol{A}$，使得（40.2）的迭代收敛比（40.1）的迭代收敛快得多.

$\boldsymbol{M}$“充分靠近 $\boldsymbol{A}$”是什么意思？本书在这一部分主要回答这一问题. 如果 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 的特征值接近于1并且 $\|\boldsymbol{M}^{-1}\boldsymbol{A}-\boldsymbol{I}\|_2$ 很小，那么已经讨论过的任何迭代都可以预期收敛很快（习题40.1). 然而，不满足如此强条件的预处理器也可以很好地做到这一点. 例如，$\boldsymbol{M}^{-1}\boldsymbol{A}$ 的特征值可以在不同于1的一个数周围聚集，并且还可以有远离其他特征值的离群特征值. 对于另一个例子，若 CGN 是迭代，那么只要求

$M^{-1}A$ 的奇异值是聚集的就足够了，而不要求 $M^{-1}A$ 的特征值是聚集的. 如通常一样，收敛速度问题的详细回答依赖于在复平面内的多项式逼近问题. 与基本迭代相反，对预处理器的分析，得到的所有改变是，我们感兴趣的是 $M^{-1}A$ 的性质而不是 A 的性质.

幸而，对于包含了不同于 CGN 的迭代的大部分问题，简单的经验法则是适用的. **如果 $M^{-1}A$ 与正规矩阵相差不大并且其特征值是聚集的，那么预处理器 M 是好的.**

40.2 左，右和埃米尔特预处理器

已经描述的内容用更精确的术语称为*左预处理器*（left preconditioner）. 另外的一个思想是把 $Ax=b$ 变换到 $AM^{-1}y=b$，其中 $x=M^{-1}y$，在此情形中，M 称为*右预处理器*（right preconditioner）. 左和右两个预处理器在实际中都使用，并且有时候两个同时使用. 为使讨论保持简单，只讨论前者.

如果 A 是埃米尔特正定矩阵，那么在预处理中通常保持这一性质. 假设 M 也是埃米尔特正定矩阵，对某一 C，$M=CC^*$. 因此（40.1）是等价于

$$[C^{-1}AC^{-*}]C^*x = C^{-1}b. \tag{40.4}$$

括号中的矩阵是埃米尔特正定的，所以这个方程可以用 CG 或相关的迭代来求解. 同时注意到，由于 $C^{-1}AC^{-*}$ 与 $C^{-*}C^{-1}A=M^{-1}A$ 相似，因此，为研究收敛性只要考察非埃米尔特矩阵 $M^{-1}A$ 的特征值就足够了. 314

40.3 例　子

图 40-1 表示对称正定矩阵预处理 CG 迭代的例子. 矩阵 A 取自习题 36.3：A 是 1000×1000 对称矩阵，其元素除了在对角线上 $a_{ij}=0.5+\sqrt{i}$，在上、下次对角线上 $a_{ij}=1$ 以及在第 100 条上、下次对角线，即对于 $|i-j|=100$，上 $a_{ij}=1$ 以外均为零. 右手边项 $b=(1,1,\cdots,1)^{T}$. 如图所示，对于这个矩阵，直接 CG 迭代收敛速度慢，在 40 次迭代之后，剩余的减少达到大约 5 个数字位. 由于矩阵是非常稀疏的，对于直接方法这是一个改进，但人们希望做得更好.

恰巧，用一个简单的对角线的预处理器就可以做得好得多. 取 $M=\mathrm{diag}(A)$，这是对角线元素为 $m_{ii}=0.5+\sqrt{i}$ 的对角矩阵. 为保持对称性，设 $C=\sqrt{M}$ 并考虑与（40.4）中一样的预处理的新迭代. 该图指出，现在迭代 30 步就给出了收敛到 15 位数字的结果.

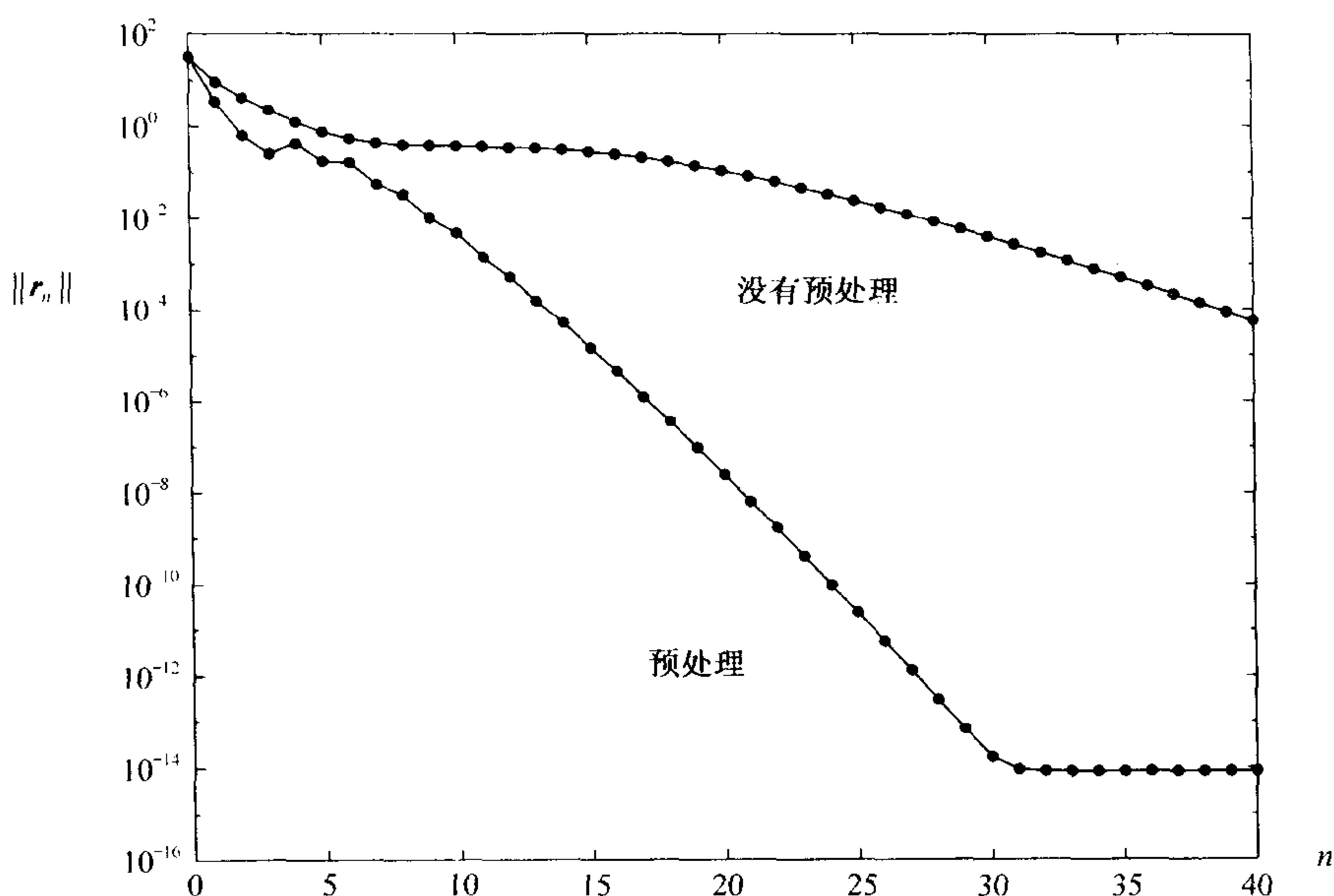


图 40-1　在正文中描述的 1000×1000 稀疏矩阵（习题 36.3 矩阵加上 0.5***I***）的 CG 和预处理 CG 的收敛曲线

315

40.4　$Ax = b$ 的预处理器述评

在实际中用的预处理器有时能像这个例子一样简单，但它们经常是更加复杂的.现在不详细地考虑一、两个例子，而是采取相反的过程，在更高水平上评述预处理思想，这些年已经证明预处理器思想非常有用. 细节可以在评注中的参考文献中找到.

对角线的缩放或雅可比式. 也许最重要的预处理器是在例子中刚提到过的一个，即$M = \text{diag}(A)$，但要以这个矩阵非奇异为条件. 对于某些问题，仅这个变换就足够使慢迭代成为快迭代. 更一般地，可以对于适当选取的向量 $c \in \mathbb{C}^m$ 取 $M = \text{diag}(c)$.要确定向量 c 使得 $\kappa(M^{-1}A)$ 被精确地极小化，是一个困难的数学问题. 但幸运的是，在实际中不需要精确的极小值，在许多情形下，如上面指出的经验法则，还有比最小化条件数更多的预处理方法.

不完全楚列斯基或 LU 因子分解. 另一个优良的预处理器使得预处理思想曾在 20 世纪 70 年代广为人知. 假设 A 是每行只有少许几个非零元素的稀疏矩阵. 如同高斯消元法或楚列斯基因子分解法中的困难一样，这些过程破坏了零，例如，若 $A = R^*R$，那么因子 R 通常不是很稀疏的. 然而，假设矩阵 $\tilde{R}$ 是用像楚列斯基的公式计算的，但允许 $\tilde{R}$ 仅在 A 有非零的位置上有非零，并且定义 $M = \tilde{R}^*\tilde{R}$. 这个不完全楚

列斯基预处理器对于某些问题可以有高度的有效性；对于不完全楚列斯基共轭梯度，常用简称 ICCG. 类似地，ILU 或*不完全 LU* 预处理器在非对称情形下是有用的. 不完全因子分解思想的许多变式已经提出并大大地发展了.

这两个预处理器的例子限于不参考原始基本问题 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 的情况. 然而，对于设计预处理器最好的建议是，仔细审视那个问题并利用其结构. 人们要问，如果在某种意义下，问题是更简单的，那么是否能很快地求解？如果是这样，那么问题更简单的变式可以是一个有效的预处理器. 大多数要举的例子是属于这一范畴的.

粗-网格 (Coarse-grid) *逼近*. 偏微分或积分方程在细网格上的离散可以产生巨大的方程组. 然而，在更粗网格上类似的离散可以产生容易求解的小的方程组. 如果能够建立一个方法，把粗网格的解转移到细网格上并且再反向转移，例如可用插值方法，那么强有力的预处理器可以由下面示意的形式得到：

$$\boldsymbol{M}=\langle\text{转移到细网格}\rangle\circ\boldsymbol{A}_{\text{粗}}\circ\langle\text{转移到粗网格}\rangle. \tag{40.5}$$

典型地，这一类型的预处理器做了处理原来问题低频分量的良好工作，留下的高频分量将由克雷洛夫子空间迭代来处理. 当这个技巧重复使用时，产生了愈来愈粗的网格序列，于是得到了*多重网格迭代* (multigrid iteration) 的思想. 316

局部逼近 (Local approximation). 粗网格逼近重视问题的某些大规模结构而忽视了某些更为精细的结构. 一种改变这个思想的类型与问题 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ 有关，其中 $\boldsymbol{A}$ 表示两个相近元素之间和互相远离元素之间的联接. 这些元素可以是像粒子这样的物理个体，或可以是在边界元素离散中引入的*板块* (panel) 这样的数值个体. 在任意情形下，考虑类似 $\boldsymbol{A}$ 的算子 $\boldsymbol{M}$ 可能是有价值的，但忽略了较长距离的交互作用而仅用 $\boldsymbol{A}$ 的近距离逼近. 在此类最简单的情况中，$\boldsymbol{M}$ 可以做对角线预处理器思想的推广，它仅仅由靠近 $\boldsymbol{A}$ 的主对角线的几条对角线所组成.

块预处理器和区域分解. 全部数值线性代数，大多数算法是根据矩阵的标量元素来表示的，该矩阵包含类似块矩阵. 对角的或雅可比预处理器可以推广到块-对角线或块-雅可比形式就是一个例子. 存在着另一类型的局部逼近，在此类型中考虑在某些分量中的局部效应而忽略此分量与其他分量的联系. 在过去的 10 年中，此类思想已经广泛地在区域分解域内推广了，在区域分解中，对于一个问题的某类子区域的解算器以灵活的方式组合来形成整体问题的预处理器. 这些方法把数学的能力与自然的可并行性结合起来了.

低阶离散. 一个微分方程或积分方程常常用高阶方法来离散，如 4 阶有限差分公式或谱方法，这样带来了精度上的利益但使得离散的模块更大并且矩阵的稀疏性减少. 具有更为稀疏矩阵的同样问题的较低阶逼近，可以成为一个有效的预处理器. 于是，例如，对于谱离散，通常碰到有限差分和有限元预处理器.

常系数或对称逼近. 如快速泊松解算器这样的特殊技巧可用于某些常系数偏微分方程. 对于变系数问题，用快速解算器实现的常系数逼近可以构成一个好的预处理器. 类似地，如果微分方程不是自共轭的，但它在某种意义下接近于非常容易求

解的自共轭方程，那么有时后者可以作为预处理器.

多项算子分裂. 许多应用包含物理效应的组合，例如，扩散和对流组合形成了流体力学的纳维-斯托克斯方程. 线性代数结果可以是矩阵问题 $\boldsymbol{Ax}=\boldsymbol{b}$，其中 $\boldsymbol{A}=\boldsymbol{A}_1+\boldsymbol{A}_2$（当然，可以有多于两项），它经常体现在非线性迭代中. 如果 $\boldsymbol{A}_1$ 或 $\boldsymbol{A}_2$ 是容易求逆的，那么它可以看作一个好的预处理器.

317

维数分裂或 ADI. 另一类分裂利用下面的事实：算子，如在二维或三维中的拉普拉斯算子，由每个分开的维中类似的算子所组成. 这个思想可以形成预处理器的基础，这一形式称为 ADI 或称为*交替方向隐式*（alternating direction implicit）方法.

经典迭代方法的一步. 在本书中没有讨论比如雅可比，高斯-赛德尔，SOR 或 SSOR 这样的“经典迭代”，但这些迭代的一步或多步（特别是雅可比和 SSOR）经常很好地作为预处理器. 这也是提出多重网格方法的关键思想之一.

周期的或卷积逼近. 纵观整个数学科学，边界条件是解析和计算困难的根源. 若仅仅没有边界条件，那么问题是在周期区域上提出的！这个思想有时可以是好的预处理器的基础. 在最简单的线性代数书中，用相关的循环矩阵 $\boldsymbol{M}(m_{ij}=m_{(i-j)(\mathrm{mod}\,m)})$，它变成了预处理一个包含特普利茨矩阵 $\boldsymbol{A}$（即 $a_{ij}=a_{i-j}$）的问题的思想，此循环矩阵 $\boldsymbol{M}$ 可以用快速傅里叶变换以 $O(m\ ln\ m)$ 次运算来求逆. 这是一个曾得到良好研究过的例子，在此例子中 $\boldsymbol{M}^{-1}\boldsymbol{A}$ 可以以范数远离单位矩阵但有高度聚集的特征值.

不稳定的直接方法. 某些数值方法，例如不选主元的高斯消元法，由于其不稳定性而产生了不精确的答案. 然而，如果不稳定方法是快的，那么为什么不用它作为预处理器呢？这是关于数值计算的“*有线传输*（fly by wire）”方法：不小心但快的求解问题. 并在稳健控制系统中嵌入那个解. 这是一个强有力的思想.

多项式预处理器. 最后，提及一个不同于其他的一个技巧，这个技巧本质上是用预处理器来逼近 $\boldsymbol{A}^{-1}$ 而不是 $\boldsymbol{A}$ 本身. 多项式预处理器是一个矩阵多项式 $\boldsymbol{M}^{-1}=\boldsymbol{p}(\boldsymbol{A})$，它具有 $\boldsymbol{p}(\boldsymbol{A})\boldsymbol{A}$ 比 $\boldsymbol{A}$ 本身有更好的迭代性质的性质. 例如，$\boldsymbol{p}(\boldsymbol{A})$ 可以从诺伊曼级数 $\boldsymbol{A}^{-1}=\boldsymbol{I}+(\boldsymbol{I}-\boldsymbol{A})+(\boldsymbol{I}-\boldsymbol{A})^2+\cdots$，或某个其他表达式的最初几项得到，它经常可由在复平面内的逼近理论得到启示. 基于用于克雷洛夫子空间迭代本身的同样的“黑箱”，实现是容易的，预处理器的系数有时可以自适应地确定.

40.5　特征值问题的预处理器

尽管这个思想最近已经发展了，但还不出名，对于特征值问题和方程组，预处理器是有效的. 在这一领域中，一些最著名的技巧类似于刚描述的方程组的多项式预处理的*多项式加速*（polynomial aueleration），*移动和倒置*（shift-and-invert）阿诺尔迪或使用 $\boldsymbol{A}$ 的有理函数代替多项式的有关的*有理克雷洛夫迭代*（rational Krylov iteration），以及基于一类对角线预处理器的戴维森和雅可比-戴维森方法. 例如，移动

318

和倒置以及有理克雷洛夫方法是基于如下事实的：如果 $r(z)$ 是一个有理函数并且 $\{\lambda_j\}$ 是 A 的特征值，那么 $r(A)$ 的特征值是 $\{r(\lambda_j)\}$. 如果 $r(A)$ 可以用合理的速度计算并且其特征值比 A 的特征值分布相对于迭代来说更好，那么这可能是快速计算特征值的方法.

40.6 结束的附注

在用预处理器这个题目来结束本书中，发现我们自己正处于未来科学计算的基本原理的中心. 计算机科学家的传统观点是计算的问题是有限的：在或短或长的计算后，可以精确地得到解. 然而，经过几年后，这个观点已变得只适合于愈来愈少的问题. 对于大规模计算的问题来说，最好的方法通常是近似方法，即在短时间内得到满意的准确的解，而不是在非常长的或无限时间内得到精确解. 数值分析在事实上首先是分析的一个分支，而不是代数的分支，甚至在求解来自线性代数的问题时亦如此. 这个现象的进一步思考在附录中作介绍.

把看来难于处理的问题变换成另一个其解可以快速近似地得到的问题的技巧，是下个世纪计算科学的中心. 对于克雷洛夫子空间矩阵迭代，这种技巧是预处理. 对于更大范围的计算问题，包含连续和离散两类，尚不清楚这个思想会带我们走向何处.

习 题

40.1 假设 $A=M-N$，其中 M 为非奇异. 假设 $\|I-M^{-1}N\|_2\leqslant\frac{1}{2}$，并且 M 被用作预处理器，如在（40.2）中.

(a) 试证明，如果把 GMRES 应用到这个预处理的问题，那么在 20 步以后，剩余范数保证小于数量的 6 阶或更好.

(b) 为得到同样的保证，CGN 需要多少步？

40.2 证明，如果矩阵 A 和预处理器 M 是埃米尔特正定的，那么不管把 M 用作左预处理器还是用作右预处理器，均可得到同样的 CG 收敛速度. 说明，这个结果为什么对于非埃米尔特矩阵和例如 GMRES，CGN 或 BCG 这样的迭代不成立.

319

附录　数值分析的定义

Lloyd N. Trefethen[1]

数值分析是什么？相信这更像是一个哲学问题．该领域内外的某些人对此持有一个错误的回答，它歪曲了处于数学科学的核心位置的数值分析的本来面目．

下面即是这个错误的回答：

$$\text{数值分析是研究舍入误差的.} \qquad (D_1)$$

读者将赞同：很难再作出一个对这领域更为乏味的描述．不错，舍入误差是不可避免的，但它们也是复杂而令人头痛的，却不是基本的．如果（D_1）是一个共同的观点，那么数值分析被普遍看作一个没有魅力的学科就不足为奇了．实际上，多年来数学家、物理学家以及计算机科学家都已习惯于认同这个对数值分析不太尊重的认识——这一看法超乎寻常地一致．

321

当然，没有人会相信或主张完全直截了当地写出（D_1）．但是考虑下面的一些标准数值分析教程的第一章标题：

Isaacson & Keller(1966)：1．范数，算法以及适定的计算．
Hamming(1971)：1．舍入误差和函数求值．
Dahlquist & Björck(1974)：1．数值计算的一些普遍原理．
2．如何求得和估计精度…….
Stoer & Bulirsch(1980)：1．误差分析．
Conte & de Boor(1980)：1．数的体系和误差．
Atkinson(1987)：1．误差：其来源，传播和分析．
Kahaner, Moler & Nash(1989)：1．引论．
2．计算机运算和计算误差．

反复出现的词为“误差”、“舍入误差”、“计算机运算”等．一个爱钻研的大学生从这些书的开始章节中会得到什么印象呢？再考虑在某些字典中的数值分析的定义．

韦氏新大学词典（1973）：“研究数学问题解的量化近似，包括误差及所包含误差的界的研究．”

钱伯斯 20 世纪词典（1983）：“研究逼近的方法及其精度，等等．”

美国传统词典（1992）：“在可能误差的范围内研究数学问题的近似解．”

1．这篇短论曾发表于 1992 年 10 月的 *SIAM News*．更早还曾发表于 1993 年 3 月/4 月的 *Bulletin of the Institute of Mathematics and Its Applications*.

又是"逼近"、"精度"、"误差"等等. 看来，这些定义非常有效地阻止了进行深入研究的好奇.

奇异值分解（SVD）提供了把数值分析理解为舍入误差科学的另一例子. 虽然 SVD 根源于 100 多年前，但主要是从 20 世纪 60 年代以来，经过 Gene Golub 以及其他数值分析工作者的工作，SVD 才达到其现在卓越的程度. 如特征值分解一样，SVD 是一个基本思想. 它是讨论包含了非对称矩阵或算子的范数以及极值的所有类型问题的自然语言. 然而，30 年后的今天，大多数数学科学家甚至很多应用数学家 [322] 都没有 SVD 的基本知识. 他们中大多数人听说过 SVD，但是对于 SVD 的普遍印象仅是消去舍入误差. 看一下一些数值分析教材就知道为什么了. 在很多情况下，SVD 是深深地隐藏在书中. 一般在一些深入的章节中，如关于秩亏损的最小二乘问题部分才会讲到，并且主要由于其稳定性质而被推荐.

可以确信，有很多人有意识或无意识地认为（D_1）至少有一半是对的. 实际上，它仅有非常小的一部分是对的. 虽然，在过去对于（D_1）的影响有历史的解释，但在今天它是一个不恰当的定义，并且可以预计到今后将更不合理.

笔者提出下面另一个定义，以此进入新世纪：

数值分析是研究连续数学问题的算法. （D_2）

各个领域之间的边界是模糊的；没有定义是完美的. 但是，对大多数学科来说，（D_2）还是提供了相对准确的描述.

关键词是*算法*. 在那些章的标题和字典的定义中，这个词在哪里？充其量也不过是把它隐藏在字里行间，但这确实是数值分析的核心：为解决某类问题而设计和分析算法.

讨论的是*连续*数学的问题. "连续"意味着包含实的或复的变量，其反面是"离散". 把许多限制条件放在一边，概括地说，数值分析与连续问题有关，而对于离散问题的算法则是其他计算机科学家所关心的.

下面考虑（D_2）的实质. 首先，由于在计算机上不能精确地表示实数和复数，所以（D_2）意味着数值分析的部分工作就是近似地表示它们，这就是产生舍入误差的原因. 现在要用有限步的算法来求解某类问题，这就是问题所在. 最重要的例子是用高斯消元法求解线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$. 为了理解高斯消元法，就必须了解诸如运算次数、机器结构等计算机科学方面的问题，并且还必须理解舍入误差的传播，即稳定性的问题. 这些就是必须明白的所有问题，如果有人断言，（D_2）只是（D_1）更为审慎的表达，那么不能用高斯消元法的例子来证明他（她）的错误.

*连续数学的大多数问题不能用有限步算法来求解！*不同于 $\boldsymbol{Ax}=\boldsymbol{b}$，也不同于计算机科学的离散问题，即使可应用精确算法来求解，数值分析的大多数问题也是不 [323] 能精确求解的. 数值分析学家知道这些，并且当他们教授计算矩阵特征值的算法时会提到这一问题，同时还会谈及阿贝尔和伽罗瓦的一些事. 但他们常常忘记去提及，同样的结论可以有效地推广到带有非线性项或导数项的任何问题——求根、求积分、

微分方程、积分方程、最优化以及可以自己指出的问题.

即使舍入误差消失了，数值分析仍会存在. 仅仅逼近数，则浮点运算的工作实际上仅是相当小的论题，并且是没有什么意义的. 数值分析更深远的工作是逼近那些未知的问题. 逼近的快速收敛是工作的目的，此领域内值得骄傲的是，对于很多问题，已经创立了收敛非常快的算法.

这些方面有时会被那些热心于符号计算的人所忽略，特别是近来的变化，使得他们往往认为 Maple 或 Mathematica 的存在使得 Matlab 和 Fortran 变得过时了. 的确，舍入误差可以在某种意义下消失：原则上，代数运算的任何有限序列可以用适当的符号运算在计算机上精确地表示. 然而，除非要求解的问题仅是有限步的，否则将不可避免地要进行长久计算以逼近最终结果，由此工作将变得非常麻烦，浮点运算是数值分析学家在计算过程中的每步进行删改的习惯而得名的，而不是指最终的单个行为. 在浮点运算或符号运算中，人们无论采用哪个方法，寻求一个快速收敛算法的主要问题是相同的.

总之，(D_2) 的一个推论是数值分析与舍入误差有关，并且也与逼近收敛相关的更深层次的误差有关，这可用不同名字（截断、离散、迭代）. 当然，可以用增加一些词汇去描述这些逼近和误差而使 (D_2) 更为明确. 但是，一旦这些词汇加了进去，就很难再停止，这是由于 (D_2) 也没有提到其他重要的问题：由于算法是在计算机上实施的，所以计算机的结构变成了问题的重要部分. 可靠性和效率是最高目标. 一些数值分析学家编写程序而另一些人则证明定理. 更重要的是所有这些工作得到了应用，每天在全球的上百万台计算机上成功地进行了无数次应用. “连续数学的问题”建立在科学与工程问题上. 没有数值方法，作为现代实践的科学与工程将很快走到尽头. 这些问题也是从牛顿时代到 20 世纪的大多数数学家所尽力研究的问题. 与任何理论数学家一样，数值分析学家是欧拉、拉格朗日、高斯和其他的伟大传统学者的继承者. 如果欧拉活到现在，那么他也不会去证明存在性定理.

324

*　*　*

如果在 10 年前，我也只会到此为止. 但在过去 10 年中，计算的发展给出了 (D_1) 和新提法 (D_2) 之间的差别.

回到 $Ax=b$. 大多数数值计算依赖于线性代数，从一开始，这一高度发展的课题就是数值分析的核心. 数值线性代数这个主题中有关当前的标准概念，例如稳定性、预处理和向后误差分析，都被精确化和深刻化了，而这些发展中的中心人物是 Jim Wilkinson，从 20 世纪 50 年代起直到 1986 年他逝世.

已经提到过 $Ax=b$ 有特殊的性质，它可以用有限步运算解出. 事实上，$Ax=b$ 有着比这更为特别的性质，对于求解它的标准算法，高斯消元法，证明有非常复杂的稳定性性质. 冯·诺依曼曾针对这个问题写过 180 页的数学文稿，图灵写了他的一篇主要文章. Wilkinson 发展了理论，由此写成了两本书并获得成功. 但事实仍是，对于某些 $n\times n$ 矩阵，带有部分选主元的高斯消元法以阶 2^n 的因子来放大误差，从而

使得这个方法在最坏的情形下变成一个无用的算法. 看来，由于具有这种性质的矩阵几乎没有，所以高斯消元法在实际中还是有用的，但到今天，还没有人能对为什么会这样作出一个令人信服的解释.[1]

那时，在各种各样的方法中，高斯消元法不是典型的方法. 很少的数值算法有如此难以捉摸的稳定性性质，当然没有人像冯·诺依曼、图灵和 Wilkinson 这样做深度的推敲. 效果怎样？高斯消元法，它应该是一个附属方法，当这个领域尚在年轻时，它在公众注意中逗留徘徊，后来才成长为数值分析学家的正规算法. 高斯消元法规定了议事日程，Wilkinson 设计了格调，不幸的结果就是（D_1）.

追究（D_1）如何得以流行的历史，当然有着比上述情形更多的内容. 在计算机的早期发展年代中，算术问题必然会得到一致的关注. 定点计算要求精密的思路和新颖的硬件；浮点计算在少许几年后作为第二次革命登场了. 直到这些事情都完全弄明白之后，很自然地，算术问题便成为数值分析的中心议题了. 而除此以外，其他的力量也在起作用. 存在着一个似乎没有名称的普遍的计算原理：*计算机愈快，算法的速度就愈重要*. 在早期年代里，使用早期的计算机，不稳定的危险性几乎同今天一样重大，并且远不为人熟悉. 然而，快速算法和慢速算法之间的差距是比较小的.

325

最近几年间出现的进展反映出我们离开那个时代已经很远了. 很多例子反复证明，即使对一个问题来说有限的算法已经存在，仍有某一无限的算法可能更好. 其差别从逻辑的观点来看似乎是绝对的，但在实际中却没有多大重要性，事实上，阿贝尔和伽罗瓦坚持认为，大规模矩阵特征值问题在实际的求解中大约与求解线性方程组一样容易. 对于求解 $\boldsymbol{Ax}=\boldsymbol{b}$，由于计算机的速度更快，矩阵的规模更巨大且较少稀疏（因 2D 模拟向 3D 的推进），迭代法现在成为愈来愈经常选取的方法，而一般的直接（=有限）算法的 $O(N^3)$ 的运算计数变得让人更加不能承受. 新策略的名称是*带有预处理的迭代法*. 不断增多的情形表明，试图以一种方法准确地求解一个问题并不是最优的，反之，先近似地求解，然后迭代. 多重网格法，或许是近 20 年来数值计算的最为重要的发展，它是基于这一思想的重复应用.

甚至直接算法也受到了计算的新方式的影响. 由于 Skeel 等人的工作，人们已注意到为使直接法稳定而付出的代价，例如高斯消元法中的选主元，在一定意义上是浪费成本的. 作为替代，跳过那一步——直接求解该问题（然而这是不稳定的）然后作一步或两步的迭代细化. “准确的”高斯消元法刚好变成了另一个预处理器.

除 $\boldsymbol{Ax}=\boldsymbol{b}$ 以外的其他一些问题也经历了类似的变化. 最著名的例子是线性规划，线性规划问题在数学上是有限的，几十年来人们用有限算法，即单纯形方法来求解. 后来 Karmarkar 于 1984 年宣称，迭代的无限算法有时会更好. 该结果引起了争论，理智的兴奋，以及整个线性规划研究领域一个可观的提升，而这个研究领域面对数

1. 这被写在第 22 讲的结论未完成之前.

值计算的主流，传统上一直处于一种不太正规的位置.

我相信对于某些问题，有限算法的存在性和其他历史因素一起，从数值分析学家的平衡观点来看，已经干扰我们几十年了. 舍入误差和不稳定性是重要的，而且数值分析学家总是这些方面的专家，他们努力避免因疏忽而犯错误. 但是，我们的主要使命是计算那些典型地不可计算的量，而且从分析观点来看要以闪电般的速度去完成. 为指导今后进一步发展，将不再研究高斯消元法及其消磨时间的稳定性性质，而应该研究神奇快速的共轭梯度迭代法，或者质点模拟的 Greengard 和 Rokhlin 的 $O(N)$ 多极算法，或者求解某些 PDE 的谱方法的指数收敛，或者对于许多类型问题的用多重网格方法在 $O(1)$ 次迭代达到收敛，或者甚至用眨眼时间就可以确定π的 1 000 000位数字的 Borwein 兄弟的令人不可思议的 AGM 迭代，那才是数值分析的核心. [326]

注：曾有许多人对本文初稿提出了意见，很难逐一列出他们的名字. 他们的建议使我知道了许多原来未知的著作.

我不认为这里表达的所有观点是完全新的. 事实上，30 年前，Peter Henrici 在他的书《数值分析初步》中把数值分析定义为“数学分析中构造方法的理论”. 其他人也曾表达了相似的观点；例如，Joseph Traub(*Communications of the ACM*, 1972)把数值分析定义为“连续算法的分析”. 对此，蓝登书屋和牛津大学出版社英语词典都提供了比这里引用的 3 个定义更好的一些定义.

再者，该领域是否应称为“数值分析”，“科学计算”，或者完全不一样的名称(“数学工程?”)，这是另一文章的内容了. [327]

注　　记

有许多关于数值线性代数的教材和论文，其中值得特别注意的是在20世纪90年代后半期出版的一些书. 在此我们不贪多求全，只列出3本流行的书，这是每个希望深入本学科的读者应该读的：

- Golub 和 Van Loan, *Matrix Computations*（矩阵计算）, 3rd ed. [GoVa96],
- Higham, *Accuracy and Stability of Numerical Algorithms*（数值算法的精度和稳定性）. [Hig96],
- Demmel, *Applied Numerical Linear Algebra*（应用数值线性代数）[Dem97].

Golub 和 Van Loan 的书，长久以来是这个领域的经典——其涉及的范围和参考书目都是广博的. Higham 的书是另一种广博的处理，它非常小心地处理细节，强调稳定性，但包含很多算法的信息和提供各种类型的见解. Demmel 的书几乎与本书有同样的题目，但在风格上完全不同，此书更集中于最近的发展和计算机总体结构的考虑，在数学基础方面略有欠缺.

关于数值线性代数的其他教材有 [Cia89], [Dat95], [GMW91], [Hag88], [Ste73], 以及 [Wat91].

还有一些优秀的教材也适用于各种更为专门的主题，包括最小二乘，特征值问题，以及迭代方法，这些将在下面的段落中一一列举出来. 对于没有包括在本书中的直接稀疏矩阵方法，两本标准教材是 [GeLi81] 和 [DER86]. 对于也未包含在本书中的软件，某些里程碑的贡献属于LAPACK[And95]以及其前导 EISPACK[Smi76] 和 LINPACK[DBMS79]，为了简化线性代数运算的代码和在特殊机器 [DDDH90] 上最大化效率，发展了基础的线性代数子程序集（BLAS），由 MathWorks, Inc.（http://www.mathworks.org）管理的 MATLAB 存储，以及到写此书时已处理过大约1.3亿次请求的 Netlib 自动软件发布系统（http://www.netlib.org）.

最后，提及非数值的线性代数材料，习惯上总是首先想到 Horn 和 Johnson 所写的两卷著名的书，[HoJo85] 和 [HoJo91].

下面转到对本书中每讲的注记上.

第1讲　矩阵-向量乘法. 不研究从矩阵行和列上运算观点出发的想法，就不可能理解20世纪数值线性代数的精髓. 实际上所有的标准算法都是规范地用这个方法表达的，虽然利用稀疏性时会进行一些修改.

原则上，对很多问题，最快的算法可以是包含子矩阵操作的递归算法，因而需要不同的思考方法. 例如，Klyuyev 和 Kokovkin-Shcherbak 在1965年指出，单独由行和列的运算解一个 $m \times m$ 方程组需要 $O(m^3)$ 次运算 [KlKo65]，但是 Strassen 和其他人的后续工作（第32讲）改进到 $O(m^{2.81})$ 甚至更低，他们采用递归到矩阵分裂为较

小块的情形 [Str69]. 计算特征值的分而治之算法（第 30 讲）是另外一个例子，其中行和列运算是不够的. 可能在下一世纪，这种算法的重要性将会成长为新的思想方法且在数值线性代数中流行，但目前还不是如此.

在 19 世纪行列式是线性代数的中心，但今天它的重要性已经降低了. 有关原因分析参见 [Axl95].

第 1 讲的材料可以在众多的教科书中找到，例如 [Str88]，形式可能有所不同. 然而，要问是否有另外的教科书对正方矩阵的维数不取 $n \times n$ 而取 $m \times m$，我们没有见到过.

第 2 讲 正交向量和矩阵. 本讲是线性代数的标准内容，在无限维情形它推广到希尔伯特空间理论中的标准内容.

基于正交矩阵的算法在 1960 年代早期随着豪斯霍尔德、Francis、吉文斯、Wilkinson、Golub 和其他人的工作成为广泛传播的方法，这些算法被公认兼具理论上的优美和数值稳定的突出性. 这个观点的快速传播可以在 Wilkinson 1965 年的专著 [Wil65] 和经典教科书 [Ste73] 以及 [LaHa95]（1974 年第一次出版）中看到.

330

第 3 讲 范数. 尽管范数的应用长期以来是泛函分析的特色，但它在线性代数中很晚才成为标准内容，甚至到今天，这些思想在非数值的线性代数教科书和课程中常常得不到重视. 对此的解释可能是线性代数的历史根源在代数而不在分析，因而在比$\mathbb{R}^m$和$\mathbb{C}^m$更一般的向量空间中才了解它们的意义. 然而，大多数的科学应用会用到实数和复数，分析和代数对它们同样有意义. 在任何带有“大小”概念的应用中，范数可能是有用的. 如果想要讨论收敛性，则一定需要它们.

在豪斯霍尔德 1964 年的书 [Hou64] 和 Forsythe 及 Moler1967 年关于高斯消元法和有关材料的简明教科书 [FoMo67] 中，范数在数值线性代数中的重要性被特别强调.

在无限维情形，像习题 3. 6 那样的对偶范数成为哈恩-巴拿赫定理 [Kat76].

第 4 和 5 讲 奇异值分解. 矩阵的 SVD 由 Beltrami(1873)和 Jordan(1874)独立地发现，并再次由 Sylvester(1889)发现. Autonne(1915)，Takagi(1925)，Williamson(1935)，Eckart 和 Young(1939)以及其他人作了相关的工作. 无限维的推广在施密特(1907) 和 Weyl(1912) 关于积分方程的论文中得到发展；参见 [Smi70]. 对于历史的讨论，参看 [HoJo91] 和 [Ste93].

尽管有这些深远的根源，SVD 在应用数学中并未被人广泛知晓，直到 20 世纪 60 年代后期 Golub 和其他作者证明了它作为很多稳定算法的基础可以有效地计算和应用. 甚至那时以后，或许因为数值分析学家专心于数值稳定性，SVD 的基本性质也是慢慢地被数学界认识. 再一次，在代数和分析之间会有不同的解释，因为使得 SVD 如此重要归根到底是它的分析性质，定理 5. 8 可以作为例证. 相反，特征值的重要性一开始就得到重视，因为特征值在本质上自然是代数的.

与 SVD 近乎相关的是极分解（polar decomposition），把矩阵表示为一个对称正

定矩阵与一个酉矩阵乘积.

在希尔伯特空间理论中，紧算子是可以用有限秩算子逼近的算子，也就是，其奇异值减少到零的算子.

第 6 讲　投影算子. 当向量展开为一组基时，就会显式或隐式地用到投影算子，正交投影算子是这样的一个算子，作为线性最小二乘法的解也是一样. 以这些材料 331 的处理为开端进行投影算子的讨论，也许是少见的，但我们也只是适度地这样做.

关于投影算子范数之间的一些关系以及互补子空间之间的角的讨论，参见 [IpMe95]. 子空间之间的角的一个完整处理一般基于 CS 分解(CS decomposition) [GoVa96].

第 7 讲　QR 因子分解. 完全和约化 QR 因子分解的差异表现在应用环节，这意味着它贯穿于数值线性代数中，但是本书没有明显地表现这个区别. QR 因子分解更通常以它的完全形式定义，而且作为应用的需要，$\boldsymbol{Q}$ 的列和 $\boldsymbol{R}$ 的行是剥离的. 这同样应用到完全及约化 SVD 之间的区别.

对于线性代数计算中矩阵因子分解重要性的重视，完全是计算机时代的产物，开始于 20 世纪 50 年代.

关于偏微分方程数值解的谱方法，参见 [CHQZ88].

第 8 讲　格拉姆-施密特正交化. 格拉姆（1883）和施密特（1907）的思想是古老且广为人知的，但它作为 QR 因子分解的解释对大多数学生来说是新的. 以我们的观点，这个解释是把一个格拉姆-施密特思想精确地固着在人们的脑海中非常宝贵的方法. 术语 QR 因子分解是由 Francis 引入的 [Fra61].

修正的格拉姆-施密特超过古典方法的优越性首先是由 Rice（1966）和 Björck（1967）建立的. 细节和参考文献在 [Bjö96] 和 [Hig96] 中给出.

画出图形来计算运算计数在优秀的教科书中不是标准的，因为同样的结果容易由代数运算获得. 但是因为我们在课堂讲授中使用了这些图形，所以我们想，为什么不把它们包括在书中?

第 9 讲　MATLAB. 到 1996 年，在数学、科学和工程的各种领域中大约已经出版了 150 种基于 MATLAB 的教科书，而且数目还在增长. 实际上全球范围的所有数值线性代数的研究者将 MATLAB 作为他们优先使用的程序语言和环境，在康奈尔大学的计算机科学系，它是所有数值分析课程的主要语言. 关于 MATLAB 的信息可以从下面获得：The MathWorks, 24 Prime Park Way, Natick, MA 01760, USA，电话 508-647-7000，传真 508-647-7001，`info@mathworks.com`，`http://www.mathworks.com`.

第 10 讲　豪斯霍尔德三角形化. 豪斯霍尔德三角形化是在 1958 年一篇经典的 4 页文章中引入的 [Hou58].（豪斯霍尔德镜射算子本身已经早在 1932 年由 Turnbull 332 和 Aitken 使用.）30 多年来，数值线性代数的研究者每 3 年开一次会议，在会议上报告他们当前的最新进展，这些会议现在被称为豪斯霍尔德专题研讨会.

为了使豪斯霍尔德镜射算子稳定，不必像我们描述的那样选择符号，替代的方法描述在［Par80］和［Hig96］中.

三角形正交化和正交三角形化之间的对称性在数学上并不是新的，但是我们后来才意识到，以前它并没有以这样简炼的形式陈述.

关于在修正格拉姆-施密特和豪斯霍尔德算法之间漂亮且令人震惊的联系，参见［BjPa92］或［Bjö96］.

第 11 讲 最小二乘问题. 最小二乘拟合思想荣誉应该归于谁？这个问题导致在数学史上高斯和勒让德之间一个优先权的大的争议，高斯在 18 世纪 90 年代发明了这个方法，而勒让德在 1805 年首先发表了它（同年高斯发明了快速傅里叶变换，他也没有发表它）. 荣誉是值得为之争取的，数学上很少有像最小二乘法这样有深远内涵的思想，但是争论给谁也没有带来荣誉，参见［Sti86］.

有些麻烦的正方形方程组，它们的解可以没有它们似乎应该有的特征，这些方程组频繁地出现在科学计算的离散化过程中. 例如，一个无限矩阵的有限部分的逆不总是收敛到无限逆矩阵相应的那一部分［Böt95］. 这种类型的困难，往往由察看替代的矩形有限矩阵，并解一个最小二乘问题来避免. 这正好是我们由图 11-1 和图 11-2顺便所做的.

Lawson 和 Hanson 的经典教科书中精彩地描绘了数值线性代数学家是怎样思考最小二乘问题的；在 1995 年的版本中，一个长长的附录综合了这本书 1974 年初版出版以来的发展［LaHa95］. 其他有价值的导引在［Str88］和［GMW91］中给出. 最近关于最小二乘问题数值方法的一个权威性的工作已由 Björck 出版［Bjö96］，读者可以由此了解此方面最新进展.

第 12 讲 条件和条件数. 矩阵条件数的思想是由阿兰·图灵于 1948 年引入的［Tur48］，同一个图灵创立了理论计算机科学，并且早在化学波在实验室被发现之前就预测了它们的可能性，在第二次世界大战期间，他参与了"Enigma"密码破译研究计划，从而结束了德国潜水艇对大西洋的控制.

关于条件的课题，一个经典的、更一般的文章是［Ric66］，也参见［Geu82］.

333 条件数的推导是扰动理论的一个特殊情形，关于矩阵和线性算子的扰动理论，权威性的参考文献是［Kat76］.

我们给出了一个简化的图，在其中我们只讨论按范数的条件数与之相对的是按分量的条件数. 后者的越来越重要的内容，参见［Hig96］和［Dem97］. 我们曾遇到的按分量思想的一个例子是 Skeel 条件数（Skeel condition number），首次在［Ske79］中给出.

在很多情形，一个适定问题的条件数反过来是与到最近不适定问题的距离有关的，至少近似地是这样. 这个观点源自 Kahan 的一篇经典的未发表的文章［Kah72］，且由 Demmel 详细地发展［Dem87］.

如在正文中指出的，例 12.4 来自 Feynman［Fey85］，可惜他没有指出他的故事

中精彩之处依赖于病态.

关于多项式根的病态在图 12-1 中作了说明，两篇最近的文章是 [EdMu95] 和 [ToTr94]，其中可以找到 Wilkinson 和其他人早期文献的启示.

对于在 n 个等距离点插值的勒贝格常数渐近于 $2^n/(e(n-1)\ln n)$ 增长的结论，是 Turetskii 在 1940 年证明的，但它并不广为人所知. 对历史的评论，参见[TrWe91].

随机矩阵是统计学家和物理学家，也是数学家的兴趣所在，习题 12.3 各部分的答案可以在 [Ede88]，[Gir90]，[Meh91] 和 [TrVi98] 中找到.

第 13 讲 浮点运算. 早在 1947 年，浮点运算就首次实现了，由此若干年来，不同制造厂商的实现细节在方法上各不相同，以至于难以保持踪迹. 这个课题由于在 20 世纪 80 年代 IEEE 标准的引入和被广泛接受而大大地简化了. 对于包含所得结果的细心讨论，参见 [Gol91] 和 [Hig96].

习题 13.3 来自 [Dem97] 的第 1 章. 一个 16 阶多项式相似的图形出现在 [Code80] 的第 3 章. Bob Lynch 告诉我们，这个例子是 Dave Dodson 作出的.

习题 13.4 的结果有时会使人惊讶，对此我们感谢 Toby Driscoll.

第 14 和 15 讲 稳定性. 向后稳定性和稳定性的概念是标准的，理论上如此，但形式上它们借助于 $O(\epsilon_{机器})$ 的精确解释来定义是不常见的. 大多数数值分析学家更喜欢抛开这些非正式的思想，以便他们能适应所需要的不同问题的特殊特征. 对于这个观点存在很好的理由，无论如何我们不认为我们已经给出的是惟一一个正确的过程. 实际上，如书中所指出的，对任意的科学计算问题，包含 $O(\epsilon_{机器})$ 的条件作为稳定性定义的基础可能太严格了.

很多像我们那样同样正式的定义可在 [deJ77] 中找到，这是一篇比它应得的影响要少的文章. 334

向后误差分析是数值分析的重大思想之一，它使得出现在本书中的数值线性代数的所有误差估计成为可能. 这个思想的提出应归功于冯·诺依曼，Goldstine，图灵，吉文斯和 Wilkinson. 近年来，向后误差分析已经由混沌动力系统的研究者重新发现，并在阴影（shadowing）的名下得到发展 [HYG88].

第 16 讲 豪斯霍尔德三角形化的稳定性. 计算出的矩阵因子 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 各自的低准确度和它们乘积的高准确度之间惊人的差异，例证了向后误差分析为什么是如此强有力的. 在 20 世纪 50 年代和 20 世纪 60 年代，Wilkinson 显示了实际出现在每个矩阵算法中的类似的效果. 第一作者足够荣幸地听取了 Wilkinson 关于这些材料的演讲，这使人惊叹有如此难忘的印象.

定理 16.1 和 16.2 是由 Wilkinson 给出的 [Wil65]，其证明也可在 [Hig96] 的 18.3 节找到. 在本书的其余部分我们不加证明地陈述了一些稳定性定理. 多数定理的证明，或者是包含证明的其他参考文献可以在 [Hig96] 中找到.

第 17 讲 回代的稳定性. 完全详尽地进行一个舍入误差分析是令人十分满意

的；某些学生认为这是本书最激励人心的一讲．结论原来是 Wilkinson 给出的，参见［Wil61］,［FoMo67］,［Hig96］.

在本讲最后的评述指出了，为什么我们更喜欢以 $O(\epsilon_{机器})$ 的方式陈述结果而不是给出明确的常数．然而，许多数值分析学家有不同的感想，包括 N. J. Higham［Hig96］，我们认为令人欣慰的是，多数情形下已经产生了明确的常数，并在出版物中记录下来了．

用到元素 ±1 的随机矩阵的习题 17.3，是基于［TrVi98］及其后的发展．

第 18 讲　最小二乘问题的条件．这个课题的文献并不特别容易阅读，部分原因是秩亏损的复杂性，这些我们已经忽略了．在这个领域的一些结果首先由 Wedin 导出［Wed73］，Stewart 的一篇文章综述了一些关键的结果［Ste77］．Stewart 和 Sun 1990 年的书进一步向前发展了，但它难以阅读［StSu90］，最近的信息最好参考［Bjö96］．文章［Geu82］和［Gra96］给出了关于弗罗贝尼斯范数准确的条件数．对于 2-范数，定理 18.1 底下的一行表示上界；就我们所知，准确的结果是不知道的．

这些条件问题的几何观点往往不是明确地描述的，但是它在［vdS75］中被强调．

335

对于稳定性分析，伪逆的微分正好是无用的；这也有算法的原因．在这方面有影响的文章是［GoPe73］.

习题 18.1 来自［GMW91］.

第 19 讲　最小二乘算法的稳定性．这是在很多书上讨论的标准材料，包括［Bjö96］,［GoVa96］和［Hig96］．列选主元的 QR 因子分解的课题属于*秩显示因子分解*（rank-revealing factorization）的一般领域内的一个大题目；参见［Bjö96］和［ChIp94］.

第 20 讲　高斯消元法．这里没有什么特别要说明的，除了将这些课题推迟到本书的中间部分．大约在 1809 年，高斯自己对正定方程组作了讨论；大约在 1857 年，雅可比把消元思想扩展到一般的矩阵．作为一个矩阵因子分解的解释，首先由Dwyer 在 1944 年发展［Dwy44］的．

第 21 讲　选主元．术语“部分”和“完全”是由 Wilkinson 在 20 世纪 50 年代提出的，但是选主元早在 1947 年已经由冯 · 诺依曼和 Goldstine 使用了．

选主元思想的众多变形在各种线性代数的计算中已经找到了应用．一个例子是*阈选主元*（threshold pivoting）的技术，在其中可以放松选主元的条件，使得主元素不必在其列中是最大的，而只要它在一个规定的最大因子范围内就可以了．虽然这样的策略会降低算法的稳定性，但它提供了额外的自由，而使得可以在稀疏矩阵的处理中用于最小化填入的排序．参见［DER86］.

第 22 讲　高斯消元法的稳定性．在 20 世纪 40 年代中期，Hotelling 和冯 · 诺依曼以及其他人预言高斯消元法一定是不稳定的，因为舍入误差指数地复合，使得方

法不能适用于几十维以上的问题. 在20世纪50年代早期，计算经验已经反映出算法终究是稳定的. 解释这种现象是一个理论上的大挑战，Wilkinson因其对这个课题的贡献而成名，他将稳定性的问题化为增长因子大小的问题. Wilkinson的分析记录在1961年的一篇里程碑式的文章［Wil61］中.

Wilkinson和他同时代的人实际没有解决为什么最坏情形的增长因子从来没有被观察到这样的问题. 在《代数特征值问题》中，他说明“经验表明，虽然这样的一个界是可以达到的，但对于实际的目的它是相当不相关的”［Wil65］，类似的评述出现在20世纪60年代到20世纪90年代的教科书中. 第一篇关于增长因子行为实质性的文章是［TrSc90］，它给出了实验的证据和实践的稳定性现象的评述完全是统计的. 本书这一讲把大增长因子和激烈变形的列空间联系起来，代表了这种统计现象在出版物上的第一个解释；更全面的分析即将到来. 336

最近Wright［Wri93］和Foster［Fos94］构造了高斯消元法不稳定的矩阵例子，事实上，虽然它们显然不会在实际计算中产生，但这样做似乎是可取的.

第23讲　楚列斯基因子分解. 楚列斯基因子分解可以用许多不同的方法描述和编程，本讲只是提供一种可能的方法. 作为利用矩阵A的一类结构的优点（正定）的方法，楚列斯基因子分解只是冰山一角. 对各类结构矩阵的方法已经设计出来了，包括对称不定、带状、箭头、范德蒙德、特普利茨、汉克尔及其他的矩阵；参见［GoVa96］.

因为技术的进步，那些使过程可能实现的精巧思想可能消失在机器内部的工作之中，只有专家才明白它们的存在. 所以对于数值算法，公众常常根本没有多大的兴趣，但它们仍然隐藏在我们所用的大多数工具之中. 习题23.3对这种现象作了一点说明. 按照惯例，一个想要解方程组的工程师会选择基于方程组性质的适当方法，而不是像MATLAB的“\”那样高水平的工具，他们更喜欢作出他们自己的决定. 到目前为止，我们仍然可以用精心的实验，得到在那些决定之下数值分析的一些进展.

第24讲　特征值问题. 尽管与在非数值的教材中强调是不同的，但这全是标准材料. 例如，讲述在计算中重要的舒尔因子分解而不讲述通常在计算中不重要的若尔当典范形，其原因在［GoWi76］中说明.

Gerschgorin的定理（习题24.1）有许多推广，其中有些在［BrRy91］和［BrMe94］中有评论.

简写“ew”和“ev”（习题24.1）是不标准的，但或许它们应该这样使用. 可以发现它们在课堂上是必不可少的.

第25讲　特征值算法综述. 虽然30多年过去了，Wilkinson的书《代数特征值问题》［Wil65］仍然是与特征值计算有关的各种类型问题细节的有价值的参考. 关于对称特征值问题，Parlett的1980年的书是标准的参考书，并且是很好的读物［Par80］. 更近的发展参见［Dem97］.

虽然没有在很多教材中提到，但习题 25.2 的迭代计数 $O(\ln(|\ln(\epsilon_{机器})|))$ 可应用到所有在科学计算中碰到的超线性收敛算法中.

第 26 讲 约化到海森伯格型或三对角型. 一个矩阵约化到海森伯格型也可以用 337 非酉运算来实现，并且渐近运算计数仅是（26.1）的一半. 原则上，非酉约化并不是经常稳定的，但在实际中它们工作得很好. 在 20 世纪 70 年代的 EISPACK 软件库 [Smi76] 中，非酉约化是作为缺省的情况推荐的，而酉约化是作为可采用的方法提供的. 在更近的 LAPACK 库 [And95] 中，仅提供了酉约化. 为什么（非酉的）高斯消元法是线性方程组的标准方法，而酉运算是特征值问题的标准方法? 虽然，酉约化对于估计特征条件数和相关的目的是方便的，但还未给出完全有力的、令人信服的回答. 这可作如下说明，鉴于既有直接方面又有迭代方面的特征值问题的较大的复杂性，数值分析工作者已不愿意拿稳定性来冒险.

要更多了解拟谱，包括计算的例子可参见 [Tre91] 和 [Tre97].

第 27 讲 瑞利商，逆迭代. 逆迭代起源于 20 世纪 40 年代的 Wielandt. 对于历史，参见[Ips97]，对于一个病态矩阵不会引起不稳定的现象的细节(习题 27.5)参见 [PeWi79]，[Par80] 或 [GoVa96].

在 20 世纪 50 年代后期，Ostrowski 所写的一系列文章中分析了瑞利商迭代的收敛性和其非对称推广 [Ost59].

计算多项式零点的一个最著名的算法是 Jenkins 和 Traub 的算法. 如在原始文章 [JeTr70] 中指出的且也在 [ToTr94] 的附录中所讨论的，Jenkins-Traub 迭代可以解释为在应用到友矩阵的瑞利商迭代中得益于稀疏性的一个格式，所以每步的工作量从 $O(m^3)$ 减少到 $O(m)$.

第 28 和 29 讲 QR 算法. QR 算法是在 1961 年由 Francis [Fra61] 和 Kublanovskaya [Kub61] 独立地基于较早的 Rutishauser 的 LR 算法而发明的，通过软件包 EISPACK [Smi76] 为世界范围使用. 此处介绍的取自于 [Wat82]. 扩充的讨论在 [Par80] 和 [Wat91] 中给出. 矩阵特征值的计算是数值分析工作者最广泛研究的问题，并且在最新的软件例如 LAPACK [And95] 中大量用到. 本书中“实用的”算法 28.2 当然不提及所有的细节，但必须加上它们对于稳健的计算. 例如，当 QR 算法在实际中执行时，利用“跟踪优势（chasing the bulge)”以更为稳定的隐式方法引入 338 位移. 见 [Par80]，[GoVa96] 或 [Dem97]，其中可以找到各种位移的性质的讨论.

第 30 讲 其他特征值算法. 雅可比关于特征值算法的主要文章出现在 1846 年 [Jac46]；他用此方法求解了与当时所知道的太阳系中 7 大行星相联系的 7×7 矩阵的特征值. 经典的现代参考文献是 [FoHe60]，包括了基于 4×4 块和 4 元的变形的更近发展可以在 [Mac95] 及其参考书中找到. 由于雅可比算法避免了三对角化步骤，所以当小心实施时，在分量方式意义下，它比 QR 算法更精确；见 [DeVe92].

分而治之算法是由 Cuppen 在 1981 年引入的 [Cup81]，并由 Dongarra 和 Sorens-

en 使其扬名天下［DoSo87］. 从那时起有关文献大增. 关于稳定性的一些关键发展以及通过快速多极方法进行加速的思想是由 Gu 和 Eisenstat 引入的；见［GuEi95］和［Dem97］.

第 31 讲 计算 SVD. 随着 Golub 和 Kahan 的文章［GoKa65］的发表，SVD 的数值计算时代始于 1965 年，此文推荐利用阶段 1 的豪斯霍尔德反射双对角化. 应用阶段 2 的 QR 算法的思想有时归功于同一文章，但事实上，在那里没有提及 QR 算法，并且在参考的 Francis［Fra61］中也没有提及. 然而，通过 Golub, Kahan, Reinsch 和 Businger 的工作，这关键思想在 20 世纪 60 年代后期非常迅速地发展起来了.

关于阶段 1 的其余方法的讨论取自［Bau94］，在其中可以找到关于奇异向量以及奇异值的细节. 关于阶段 2 的信息见［GoVa96］和［Dem97］.

第 32 讲 迭代法综述. 克雷洛夫子空间迭代方法出现的历史是极有趣的. 其基础在 20 世纪 50 年代初期就有了，但是，对于这些方法成为大多数问题的优良方法的那个时代，机器委实太慢了. 很自然地，不仅它们没有广泛使用，而且它们关于渐近复杂性的主要长处也没有被清楚理解. 现在，人们会自动地注意到算法的渐近复杂性；在 20 世纪 50 年代并不如此.

另一方面，某些“经典的迭代”，比如高斯-赛德尔和 SOR 是在 20 世纪 50 年代对由偏微分方程离散产生的问题广泛使用的. 由于这些方法在很多书中已有描述，然而现在的实际重要性在减少，所以在此没有给予注意. 关于此题目的经典参考文献是［Var62］.

关于稀疏直接矩阵算法，见［GeLi81］和［DER86］.

作为时间的函数，一个“大的”矩阵的维数 m 是什么？在最近几年内关于此主题的信息已经由 Edelman 收集起来，例如，他在 1994 年报告，虽然 $m = 76\ 800$ 的矩阵已经处理了［Ede94］，但他还不知道 $m > 100\ 000$ 的稠密方程组的任何解法. 339

最近已经写出了许多关于迭代方法的书；特别推荐的是由 Saad 所写的关于特征值［Saa92］和关于线性方程组［Saa96］的专著，以及由 Greenbaum 所写的即将出版的关于线性方程组的教材［Gre97］. 关于这个题目的其他书包括，有关预处理的［Axe94］，强调向非线性问题推广的［Kel95］以及［Bru95］,［Fis96］,［Hac94］和［Wei96］.

从 20 世纪 50 年代以来，已经认识到克雷洛夫子空间方法不仅可以应用到矩阵，而且可应用到线性算子. 在这个趋势中较早的参考文献是［Dan71］，最近的进展是［Nev93］.

投影到低维子空间上的克雷洛夫思想似乎类似于一个数值计算的中心思想——连续问题的离散，因而它变成有限维的. 人们可能问，是否它比一个类似物有更多的内容，如果是这样，是否有可能把离散和迭代组成到一个过程而不是分别地用 m 代替 ∞（离散）和用 n 代替 m（迭代）. 这个回答，至少在某些情况中，当然是对的. 然而，这种类型的很多可能性还没有被利用，并且在当前，大多数科学计算仍

保持在把离散和迭代分开的状态.

Strassen 的著名文章出现在 1969 年［Str69］，对于在图 32-2 中表示的较低指数的算法的启示可以在［Pan84］和［Hig96］中找到. 现在最好的指数为 2.376，归功于 Coppersmith 和 Winograd［CoWi90］.

在本书中曾将其称之为“计算机科学的基本定律”的那个原理，它的名称一般并不为人所知. 该原理在［AHU74］中讨论；但未知在何处首先发表.

第 33 讲　阿诺尔迪迭代. 阿诺尔迪的原始文章写于 1951 年，但其内涵距现在相当远［Arn51］. 认识到阿诺尔迪、兰乔斯、CG 以及其他方法之间的联系花了很长时间.

第 34 讲　用阿诺尔迪迭代求特征值. 兰乔斯迭代的收敛性已被相当好地理解. 一些关键文章是 Kaniel［Kan66］,Paige［Pai71］和 Saad［Saa80］. 然而，更一般的阿诺尔迪迭代的收敛性未被完全理解. 有些结果是可用的，见［Saa92］.

根据双纽线的讨论是非标准的. 在［GrTr94］中发展了包括理想的阿诺尔迪和 GMRES 多项式概念的与多项式逼近的联系. 计算基于半定的程序设计的这些多项式的算法以及对于拟谱相关双纽线的例子出现在［ToTr98］中. 通过阿诺尔迪迭代来估计拟谱的思想来自［ToTr96］.

关于“警告的注解”见［TTRD93］,［Tre91］和［Tre97］.

340

第 35 讲　GMRES. 尽管各种与 GMRES 算法相关的算法早已出现了，但 GMRES 算法最近由 Saad 和 Schultz 于 1986 年意外地提出来［SaSc86］.

第 36 讲　兰乔斯迭代. 兰乔斯迭代开始于 1950 年［Lan50］. 尽管它与共轭梯度法紧密相关，但它是独立地表达出来的. 当易于处理的矩阵问题增加其规模时，兰乔斯迭代变得与其他方法不相上下，于是在 20 世纪 70 年代它被“重新发现了”. 两卷本处理是由 Cullum 和 Willoughby 在 1985 年给出的［CuWi85］.

经过多项式逼近完全建立了克罗洛夫子空间迭代与位势理论（电荷）之间的联系. 从位势理论能否推导出收敛的详细分析见［DTT98］.

第 37 讲　从兰乔斯到高斯求积. 自 1969 年以来，人们已经认识到计算高斯求积的节点和权的正确方法是通过三对角矩阵的特征值问题［GoWe69］. 这里简单的表示完全描述了除了一个省略点的联系：权与可以从克里斯托费尔-达布公式导出的特征向量的第一个分量的关系. 对于正交多项式的这个问题和其他材料的信息，经典的参考文献是由 Szegö 所写的书［Sze75］.

注意到对于大的 n，第 n 阶牛顿-科茨公式有 2^n 阶系数. 当牛顿-科茨公式用插值来推导时，这基本上与在第 12 讲的注释中提到的与勒贝格常数有关的因子 2^n 相同.

第 38 讲　共轭梯度法. 共轭梯度法迭代独立地源自 Hestenes 和 Stiefel，但是为了这个主要论文两人很早就建立了联系（1951 年 8 月），他们的文章是数值分析领域

的经典成果，可以说是合作的产物［HeSt52］. 像兰乔斯迭代一样，CG 是在 20 世纪 70 年代"重新发现的"，并且立刻成为科学计算的主要支柱. 关于 CG 和兰乔斯迭代紧密地交织在一起的历史见［GoOL89］.

在浮点运算中 CG 迭代性质的大部分知识属于 Greenbaum 和她的合作者；见［Gre97］.

第 39 讲　双正交化方法. 双共轭梯度迭代始于 1952 年的兰乔斯［Lan52］并由 Fletcher 于 1976 使其再度流行（并重新命名）［Fle76］. 在教材中提及的另一方法是向前看兰乔斯［PTL85］，CGS［Son89］，QMR［FrNa91］，Bi-CGSTAB［vdV92］和 TFQMR［Fre93］. 到 1991 年为止的综述，见［FGN92］，这些算法与正交多项式、连分式、Padè 逼近以及其他课题的深刻联系的描述见［Gut92］，其他有价值的参考文献是［Saa96］.

对矩阵性质（决定各种类型的非对称矩阵迭代的收敛性）的比较，见［NRT92］，其中习题 39.1 和 39.2 也是取自其中. 对于 BCG 和 QMR 之间关系的特殊讨论，见［FrNa91］和［CuGr96］，其中指出了在 BCG 收敛曲线中的峰值以精确方式对应于 QMR 收敛曲线中的平坦（慢过程）部分. 341

第 40 讲　预处理. 单词"预处理"始自 1948 年的图灵，在矩阵迭代范围内的一些早期贡献属于 Hestenes，Engeli，Wachspress，Evans 和 Axelsson. 这个思想在 20 世纪 70 年代由 Meijerink 和 van der Vorst［Meva77］引入了不完全因子分解后而变得著名，在那个年代另一篇有影响的文章是［CGO76］. 这个技巧的当前情况总结可推荐阅读［Axe94］和［Saa96］. 区域分解在［SBG96］中讨论，用不稳定的直接方法作为预处理器是在［Ske80］中讨论的. 特普利茨矩阵的循环预处理器的思想由 Strang［Str86］首创，并且其后已经大大地推广了.

正如瑞利商位移加速逆迭代由线性加速到 3 次收敛一样，用在每步自适应地改变预处理器加速一个迭代将有什么结果? 这个思想是有前途的，并且最近引起了一些注意，见［Saa96］.

关于特征值问题的预处理器，虽然戴维森的原始文章出自 1975 年的［Dav75］，但在 20 世纪 90 年代才获得了应有的信誉；这些方法的一个好的导引文献是［Saa92］. 多项式加速设计已由 Chatelin［Cha93］，Saad，Scott，Lehoucq 和 Sorensen［LeSo96］和其他作者发展了. 位移和反转阿诺尔迪方法已由 Saad 和 Spence 发展了，有理克雷洛夫迭代已由 Ruhe 发展了；关于最近的综述见［MeRo96］. Sleijpen 和 van der Vorst［Slvd96］引入了雅可比-戴维森算法. 342

参考文献

[AHU74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[And95] E. Anderson et al., *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia, 1995.

[Arn51] W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math. 9 (1951), 17–29.

[Axe94] O. Axelsson, *Iterative Solution Methods*, Cambridge U. Press, Cambridge, UK, 1994.

[Axl95] S. Axler, *Down with determinants*, Amer. Math. Monthly 102 (1995), 139–154.

[Bar94] R. Barrett et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, 1994.

[Bau94] D. Bau, *Faster SVD for matrices with small* m/n, TR94-1414, Computer Science Dept., Cornell U., 1994.

[Bjö96] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[BjPa92] Å. Björck and C. C. Paige, *Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm*, SIAM J. Matrix Anal. Appl. 13 (1992), 176–190.

[Böt95] A. Böttcher, *Infinite matrices and projection methods*, in P. Lancaster, ed., Lectures on Operator Theory and Its Applications, Amer. Math. Soc., Providence, RI, 1995.

[BrMe94] R. A. Brualdi and S. Mellendorf, *Regions in the complex plane containing the eigenvalues of a matrix*, Amer. Math. Monthly 101 (1994), 975–985.

[BrRy91] R. A. Brualdi and H. J. Ryser, *Combinatorial Matrix Theory*, Cambridge U. Press, Cambridge, UK, 1991.

[Bru95] A. M. Bruaset, *A Survey of Preconditioned Iterative Methods*, Addison-Wesley Longman, Harlow, Essex, UK, 1992.

[CHQZ88] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York, 1988.

[ChIp94] S. Chandrasekaran and I. C. F. Ipsen, *On rank-revealing factorisations*, SIAM J. Matrix Anal. Appl. 15 (1994), 592–622.

[Cha93] F. Chatelin, *Eigenvalues of Matrices*, Wiley, New York, 1993.

[Cia89] P. G. Ciarlet, *Introduction to Numerical Linear Algebra and Optimisation*, Cambridge U. Press, Cambridge, UK, 1989.

[CGO76] P. Concus, G. H. Golub, and D. P. O'Leary, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in J. R. Bunch and D. J. Rose, eds., Sparse Matrix Computations, Academic Press, New York, 1976.

[Code80] S. D. Conte and C. de Boor, *Elementary Numerical Analysis: An Algorithmic Approach*, 3rd ed., McGraw-Hill, New York, 1980.

[CoWi90] D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, J. Symbolic Comput. 9 (1990), 251–280.

[CuGr96] J. Cullum and A. Greenbaum, *Relations between Galerkin and norm-minimizing iterative methods for solving linear systems*, SIAM J. Matrix Anal. Appl. 17 (1996), 223–247.

[CuWi85] J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, v.* 1 *and* 2, Birkhäuser, Boston, 1985.

[Cup81] J. J. M. Cuppen, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math. 36 (1981), 177–195.

[Dan71] J. W. Daniel, *The Approximate Minimization of Functionals*, Prentice Hall, Englewood Cliffs, NJ, 1971.

[Dat95] B. N. Datta, *Numerical Linear Algebra and Applications*, Brooks/Cole, Pacific Grove, CA, 1995.

[Dav75] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices*, J. Comp. Phys. 17 (1975), 87–94.

[deJ77] L. S. de Jong, *Towards a formal definition of numerical stability*, Numer. Math. 28 (1977), 211–219.

[Dem87] J. W. Demmel, *On condition numbers and the distance to the nearest ill-posed problem*, Numer. Math. 51 (1987), 251–289.

[Dem97] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[DeVe92] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl. 13 (1992), 1204–1245.

[DBMS79] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *LINPACK Users' Guide*, SIAM, Philadelphia, 1979.

[DDDH90] J. J. Dongarra, J. J. Du Croz, I. S. Duff, and S. J. Hammarling, *Algorithm* 679. *A set of level* 3 *basic linear algebra subprograms: Model implementation and test programs*, ACM Trans. Math. Software 16 (1990), 18–28.

[DoSo88] J. J. Dongarra and D. C. Sorensen, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Stat. Comput. 8 (1987), s139–s154.

[DTT98] T. A. Driscoll, K.-C. Toh, and L. N. Trefethen, *From potential theory to matrix iterations in six steps*, SIAM Review 40 (1998), 547–578.

[DER86] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, UK, 1986.

[Dwy44] P. S. Dwyer, *A matrix presentation of least squares and correlation theory with matrix justification of improved methods of solutions*, Ann. Math. Stat. 15 (1944), 82–89.

[Ede88] A. Edelman, *Eigenvalues and condition numbers of random matrices*, SIAM J. Matrix Anal. Appl. 9 (1988), 543–560.

[Ede94] A. Edelman, *Large numerical linear algebra in* 1994*: The continuing influence of parallel computing*, Proc. 1994 Scalable High Performance Computing Conf., IEEE Computer Soc. Press, Los Alamitos, CA, 1994, 781–787.

[EdMu95] A. Edelman and H. Murakami, *Polynomial roots from companion matrix eigenvalues*, Math. Comp. 64 (1995), 763–776.

[Fey85] R. P. Feynman, *Surely You're Joking, Mr. Feynman! Adventures of a Curious Character*, Norton, New York, 1985.

[Fis96] B. Fischer, *Polynomial Based Iteration Methods for Symmetric Linear Systems*, Wiley-Teubner, Chichester, UK, 1996.

[Fle76] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in G. A. Watson, ed., Numerical Analysis Dundee 1975, Lec. Notes in Math. v. 506, Springer-Verlag, Berlin, 1976, 73–89.

[FoHe60] G. E Forsythe and P. Henrici, *The cyclic Jacobi method for computing the principal values of a complex matrix*, Trans. Amer. Math. Soc. 94 (1960), 1–23.

[FoMo67] G. E. Forsythe and C. B. Moler, *Computer Solution of Linear Algebraic Systems*, Prentice Hall, Englewood Cliffs, NJ, 1967.

[Fos94] L. V. Foster, *Gaussian elimination with partial pivoting can fail in practice*, SIAM J. Matrix Anal. Appl. 15 (1994), 1354–1362.

[Fra61] J. G. F. Francis, *The QR transformation: A unitary analogue to the LR transformation, parts* I *and* II, Computer J. 4 (1961), 256–72 and 332–45.

[Fre93] R. W. Freund, *A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems*, SIAM J. Sci. Stat. Comput. 13 (1992), 425–448.

[FGN92] R. W. Freund, G. H. Golub, and N. M. Nachtigal, *Iterative solution of linear systems*, Acta Numerica 1 (1992), 57–100.

[FrNa91] R. W. Freund and N. M. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math. 60 (1991), 315–339.

[GeLi81] A. George and J. W.-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice Hall, Englewood Cliffs, NJ, 1981.

[Geu82] A. J. Geurts, *A contribution to the theory of condition*, Numer. Math. 39 (1982), 85–96.

[GMW91] P. E. Gill, W. Murray, and M. H. Wright, *Numerical Linear Algebra and Optimization*, Addison-Wesley, Redwood City, CA, 1991.

[Gir90] V. L. Girko, *Theory of Random Determinants*, Kluwer, Dordrecht, the Netherlands, 1990.

[Gol91] D. Goldberg, *What every computer scientist should know about floating-point arithmetic*, ACM Computing Surveys 23 (1991), 5–48.

[GoKa65] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal. 2 (1965), 205–224.

[GoOL89] G. H. Golub and D. P. O'Leary, *Some history of the conjugate gradient and Lanczos methods*, SIAM Review 31 (1989), 50–100.

[GoPe73] G. H. Golub and V. Pereyra, *The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate*, SIAM J. Numer. Anal. 10 (1973), 413–432.

[GoVa96] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed.,

Johns Hopkins U. Press, Baltimore, 1996.

[GoWe69] G. H. Golub and J. H. Welsch, *Calculation of Gauss quadrature rules*, Math. Comp. 23 (1969), 221–230.

[GoWi76] G. H. Golub and J. H. Wilkinson, *Ill-conditioned eigensystems and the computation of the Jordan canonical form*, SIAM Review 18 (1976), 578–619.

[Gra96] S. Gratton, *On the condition number of linear least squares problems in a weighted Frobenius norm*, BIT 36 (1996), 523–530.

[Gre97] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.

[GrTr94] A. Greenbaum and L. N. Trefethen, *GMRES/CR and Arnoldi/Lanczos as matrix approximation problems*, SIAM J. Sci. Comput. 15 (1994), 359–368.

[GuEi95] M. Gu and S. C. Eisenstat, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl. 16 (1995), 172–191.

[Gut92] M. H. Gutknecht, *A completed theory of the unsymmetric Lanczos process and related algorithms, part* I, SIAM J. Matrix Anal. Appl. 13 (1992), 594–639.

[Hac94] W. Hackbusch, *Iterative Solution of Large Sparse Linear Systems of Equations*, Springer-Verlag, Berlin, 1994.

[Hag88] W. Hager, *Applied Numerical Linear Algebra*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[HYG88] S. M. Hammel, J. A. Yorke, and C. Grebogi, *Numerical orbits of chaotic processes represent true orbits*, Bull. Amer. Math. Soc. 19 (1988), 465–469.

[HeSt52] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand. 49 (1952), 409–436.

[Hig96] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[HoJo85] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge U. Press, Cambridge, UK, 1985.

[HoJo91] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge U. Press, Cambridge, UK, 1991.

[Hou58] A. S. Householder, *Unitary triangularization of a nonsymmetric matrix*, J. Assoc. Comput. Mach. 5 (1958), 339–342.

[Hou64] A. S. Householder, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, 1964.

[Ips97] I. C. F. Ipsen, *A history of inverse iteration*, in B. Huppert and H. Schneider, eds., Helmut Wielandt, Mathematische Werke, Mathematical Works, v. 2, Walter de Gruyter, Berlin, 1996, 453–463.

[IpMe95] I. C. F. Ipsen and C. D. Meyer, *The angle between complementary subspaces*, Amer. Math. Monthly 102 (1995), 904–911.

[Jac46] C. G. J. Jacobi, *Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*, J. Reine Angew. Math. 30 (1846), 51–94.

[JeTr70] M. A. Jenkins and J. F. Traub, *A three-stage variable-shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration*, Numer. Math. 14 (1970), 252–263.

[Kah72] W. M. Kahan, *Conserving confluence curbs ill-condition*, unpublished manuscript, 1972.

[Kan66] S. Kaniel, *Estimates for some computational techniques in linear algebra*, Math. Comp. 20 (1966), 369–378.

[Kat76] T. Kato, *Perturbation Theory for Linear Operators*, 2nd ed., Springer-Verlag, New York, 1976.

[Kel95] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.

[KlKo65] V. V. Klyuyev and N. I. Kokovkin-Shcherbak, *On the minimization of the number of arithmetic operations for the solution of linear algebraic systems of equations*, Zh. Vychisl. Mat. i Mat. Fiz. 5 (1965), 21–33; translated from the Russian by G. J. Tee, Tech. Rep. CS24, Computer Science Dept., Stanford University, 1965.

[Koz92] D. C. Kozen, *The Design and Analysis of Algorithms*, Springer-Verlag, New York, 1992.

[Kry31] A. N. Krylov, *On the numerical solution of equations which in technical questions are determined by the frequency of small vibrations of material systems*, Izv. Akad. Nauk. S. S. S. R. Otd Mat. Estest. 1 (1931), 491–539.

[Kub61] V. N. Kublanovskaya, *On some algorithms for the solution of the complete eigenvalue problem*, USSR Comp. Math. Phys. 3 (1961), 637–657.

[Lan50] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Stand. 45 (1950), 255–282.

[Lan52] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bur. Stand. 49 (1952), 33–53.

[LaHa95] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, SIAM, Philadelphia, 1995 (reprinting with corrections and a new appendix of a 1974 Prentice Hall text).

[LeSo96] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl. 17 (1996), 789–821.

[Mac95] N. Mackey, *Hamilton and Jacobi meet again: Quaternions and the eigenvalue problem*, SIAM J. Matrix Anal. Appl. 16 (1995), 421–435.

[MeRo96] K. Meerbergen and D. Roose, *Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems*, IMA J. Numer. Anal. 16 (1996), 297–346.

[Meh91] M. L. Mehta, *Random Matrices*, 2nd ed., Academic Press, San Diego, 1991.

[Meva77] J. Meijerink and H. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp. 31 (1977), 148–162.

[NRT92] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl. 13 (1992), 778–795.

[Nev93] O. Nevanlinna, *Convergence of Iterations for Linear Equations*, Birkhäuser, Basel, 1993.

[Ost59] A. M. Ostrowski, *On the convergence of the Rayleigh quotient iteration for the computation of characteristic roots and vectors,* IV. *Generalized Rayleigh quotient for nonlinear elementary divisors*, Arch. Rational Mech. Anal. 3 (1959), 341–347.

[Pai71] C. C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, PhD diss., U. of London, 1971.

[Pan84] V. Pan, *How to Multiply Matrices Faster*, Lec. Notes in Comp. Sci., v. 179, Springer-Verlag, Berlin, 1984.

[Par80] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[PTL85] B. N. Parlett, D. R. Taylor, and Z. A. Liu, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Math. Comp. 44 (1985), 105–124.

[PeWi79] G. Peters and J. H. Wilkinson, *Inverse iteration, ill-conditioned equations and Newton's method*, SIAM Review 21 (1979), 339–360.

[Ric66] J. R. Rice, *A theory of condition*, SIAM J. Numer. Anal. 3 (1966), 287–310.

[Saa80] Y. Saad, *On the rates of convergence of the Lanczos and the block Lanczos methods*, SIAM J. Numer. Anal. 17 (1980), 687–706.

[Saa92] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester U. Press, Manchester, UK, 1992.

[Saa96] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.

[SaSc86] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. 7 (1986), 856–869.

[Ske79] R. D. Skeel, *Scaling for numerical stability in Gaussian elimination*, J. Assoc. Comput. Mach. 26 (1979), 494–526.

[Ske80] R. D. Skeel, *Iterative refinement implies numerical stability for Gaussian elimination*, Math. Comp. 35 (1980), 817–832.

[Slvd96] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl. 17 (1996), 401–425.

[Smi76] B. T. Smith et al., *Matrix Eigensystem Routines—EISPACK Guide*, Springer-Verlag, Berlin, 1976.

[SBG96] B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge U. Press, Cambridge, UK, 1996.

[Smi70] F. Smithies, *Integral Equations*, Cambridge U. Press, Cambridge, UK, 1970.

[Son89] P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. 10 (1989), 36–52.

[Ste73] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[Ste77] G. W. Stewart, *On the perturbation of pseudo-inverses, projections, and linear least squares problems*, SIAM Review 19 (1977), 634–662.

[Ste93] G. W. Stewart, *On the early history of the singular value decomposition*, SIAM Review 35 (1993), 551–566.

[StSu90] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[Sti86] S. M. Stigler, *The History of Statistics*, Harvard U. Press, Cambridge, MA, 1986.

[Str86] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math. 74 (1986), 171–176.

[Str88] G. Strang, *Linear Algebra and Its Applications*, 3rd ed., Harcourt, Brace, and Jovanovich, San Diego, 1988.

[Str69] V. Strassen, *Gaussian elimination is not optimal*, Numer. Math. 13 (1969), 354–356.

[Sze75] G. Szegő, *Orthogonal Polynomials*, 4th ed., Amer. Math. Soc., Providence, RI, 1975.

[ToTr94] K.-C. Toh and L. N. Trefethen, *Pseudozeros of polynomials and pseudospectra of companion matrices*, Numer. Math. 68 (1994), 403–425.

[ToTr96] K.-C. Toh and L. N. Trefethen, *Computation of pseudospectra by the Arnoldi iteration*, SIAM J. Sci. Comput. 17 (1996), 1–15.

[ToTr98] K.-C. Toh and L. N. Trefethen, *The Chebyshev polynomials of a matrix*, SIAM J. Matrix Anal. Appl. 20 (1998), 400–419.

[Tre91] L. N. Trefethen, *Pseudospectra of matrices*, in D. F. Griffiths and G. A. Watson, eds., Numerical Analysis 1991, Longman Scientific and Technical, Harlow, Essex, UK, 1992, 234–266.

[Tre97] L. N. Trefethen, *Pseudospectra of linear operators*, SIAM Review 39 (1997), 383–406.

[TrSc90] L. N. Trefethen and R. S. Schreiber, *Average-case stability of Gaussian elimination*, SIAM J. Matrix Anal. Appl. 11 (1990), 335–360.

[TTRD93] L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll, *Hydrodynamic stability without eigenvalues*, Science 261 (1993), 578–584.

[TrVi98] L. N. Trefethen and D. Viswanath, *Condition numbers of random triangular matrices*, SIAM J. Matrix Anal. Appl. 19 (1998), 565–581.

[TrWe91] L. N. Trefethen and J. A. C. Weideman, *Two results on polynomial interpolation in equally spaced points*, J. Approx. Theory 65 (1991), 247–260.

[Tur48] A. M. Turing, *Rounding-off errors in matrix processes*, Quart. J. Mech. Appl. Math. 1 (1948), 287–308.

[vdS75] A. van der Sluis, *Stability of the solutions of linear least squares problems*, Numer. Math. 23 (1975), 241–254.

[vdV92] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly convergent variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. 13 (1992), 631–644.

[Var62] R. S. Varga, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1962.

[Wat82] D. S. Watkins, *Understanding the QR algorithm*, SIAM Review 24 (1982), 427–440.

[Wat91] D. S. Watkins, *Fundamentals of Matrix Computations*, Wiley, New York, 1991.

[Wed73] P.-Å. Wedin, *Perturbation theory for pseudo-inverses*, BIT 13 (1973), 217–232.

[Wei96] R. Weiss, *Parameter-Free Iterative Linear Solvers*, Akademie Verlag, Berlin, 1996.

[Wil61] J. H. Wilkinson, *Error analysis of direct methods of matrix inversion*, J. Assoc. Comput. Mach. 8 (1961), 281–330.

[Wil65] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

[Wri93] S. J. Wright, *A collection of problems for which Gaussian elimination with partial pivoting is unstable*, SIAM J. Sci. Comput. 14 (1993), 231–238.

索　引

索引中页码为英文原书页码，与书中页边标注的页码一致.

D

H

I

N

O

P

Q